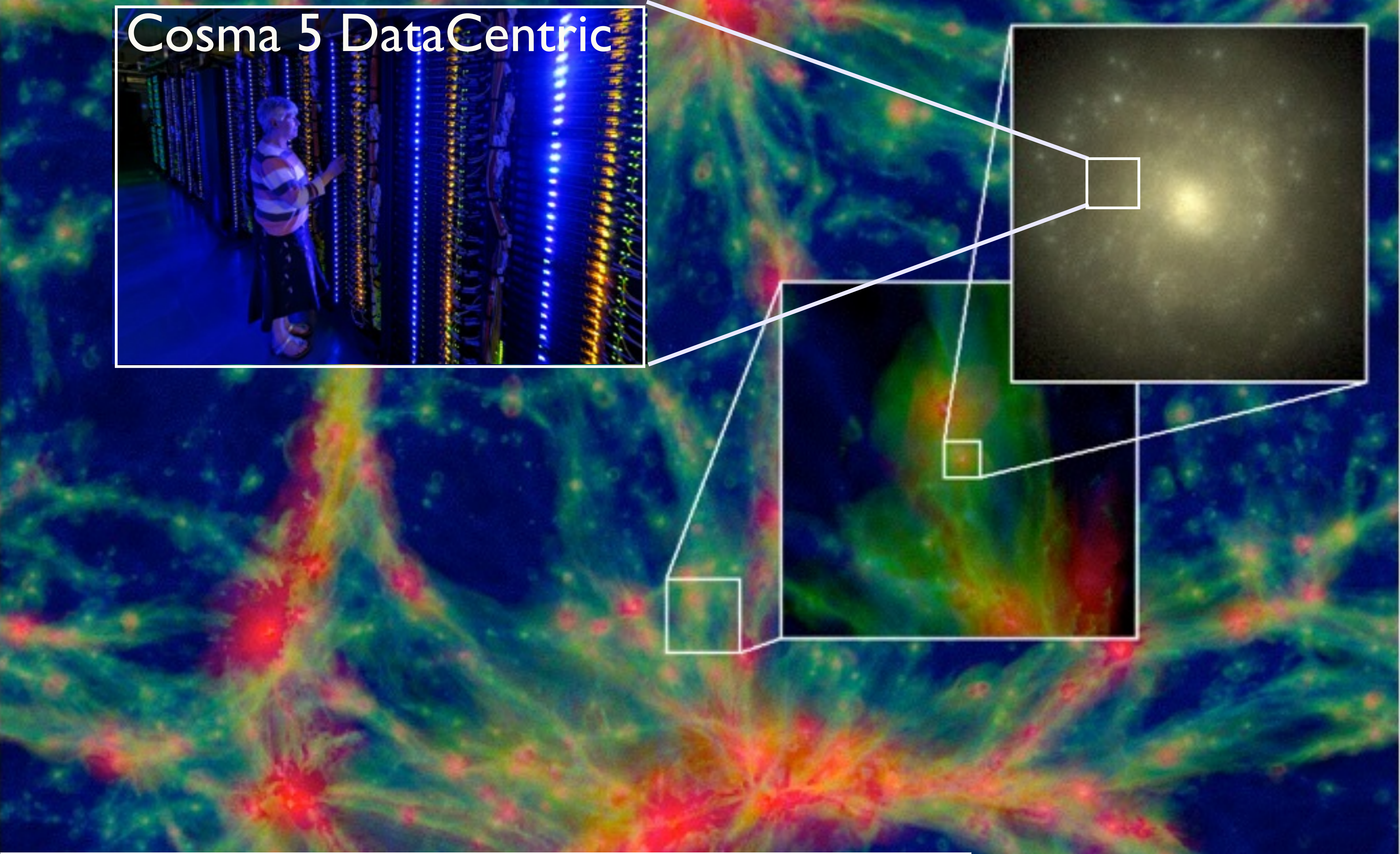
The Swift simulation code

Tom Theuns
ICC, Durham

Tom Theuns: Intro to SPH
Bert VandenBroucke: GIZMO
Pedro Gonnet: Task-based parallelism
Matthieu Schaller: Swift

Cosmological hydrodynamical simulations

Eagle in Durham

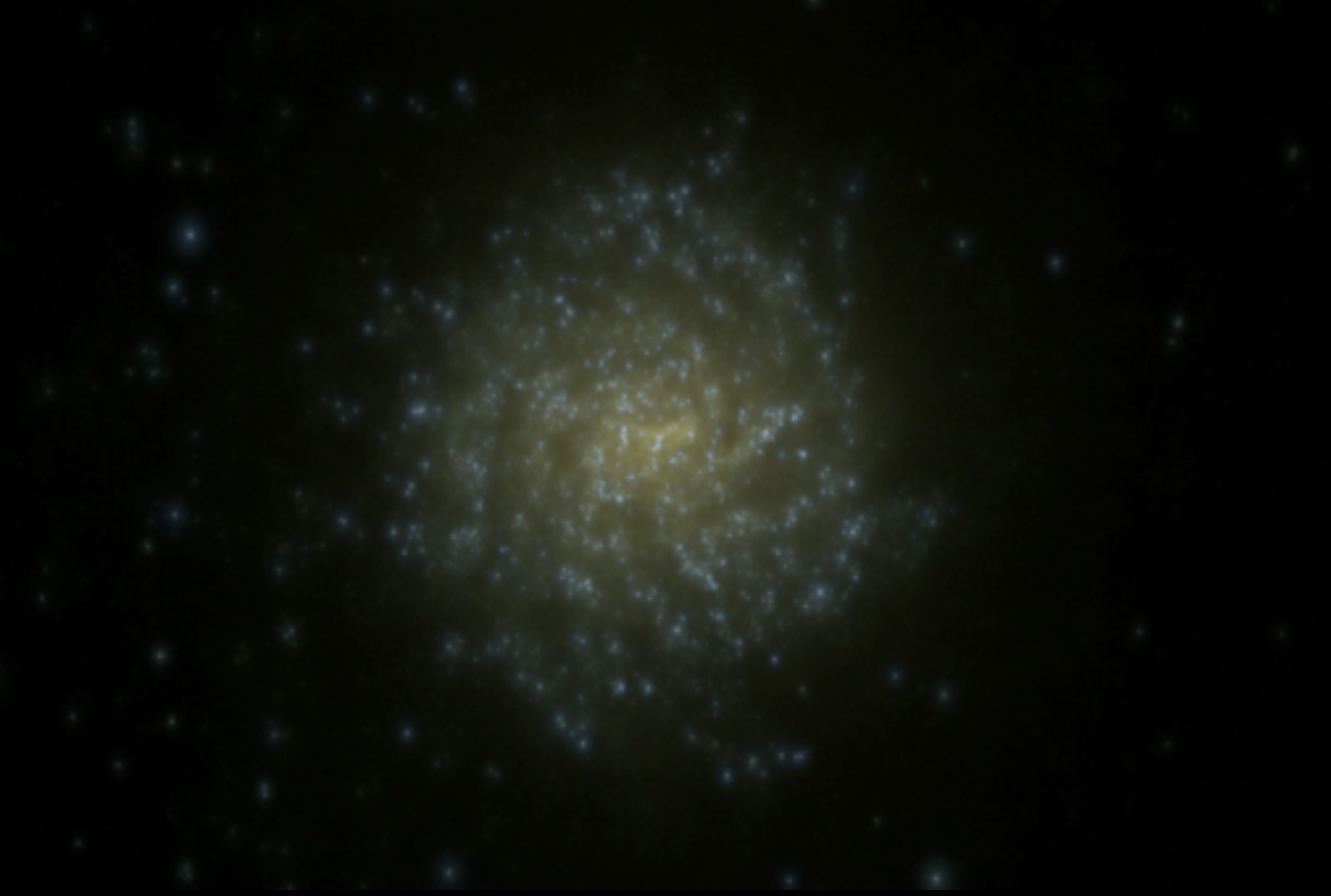


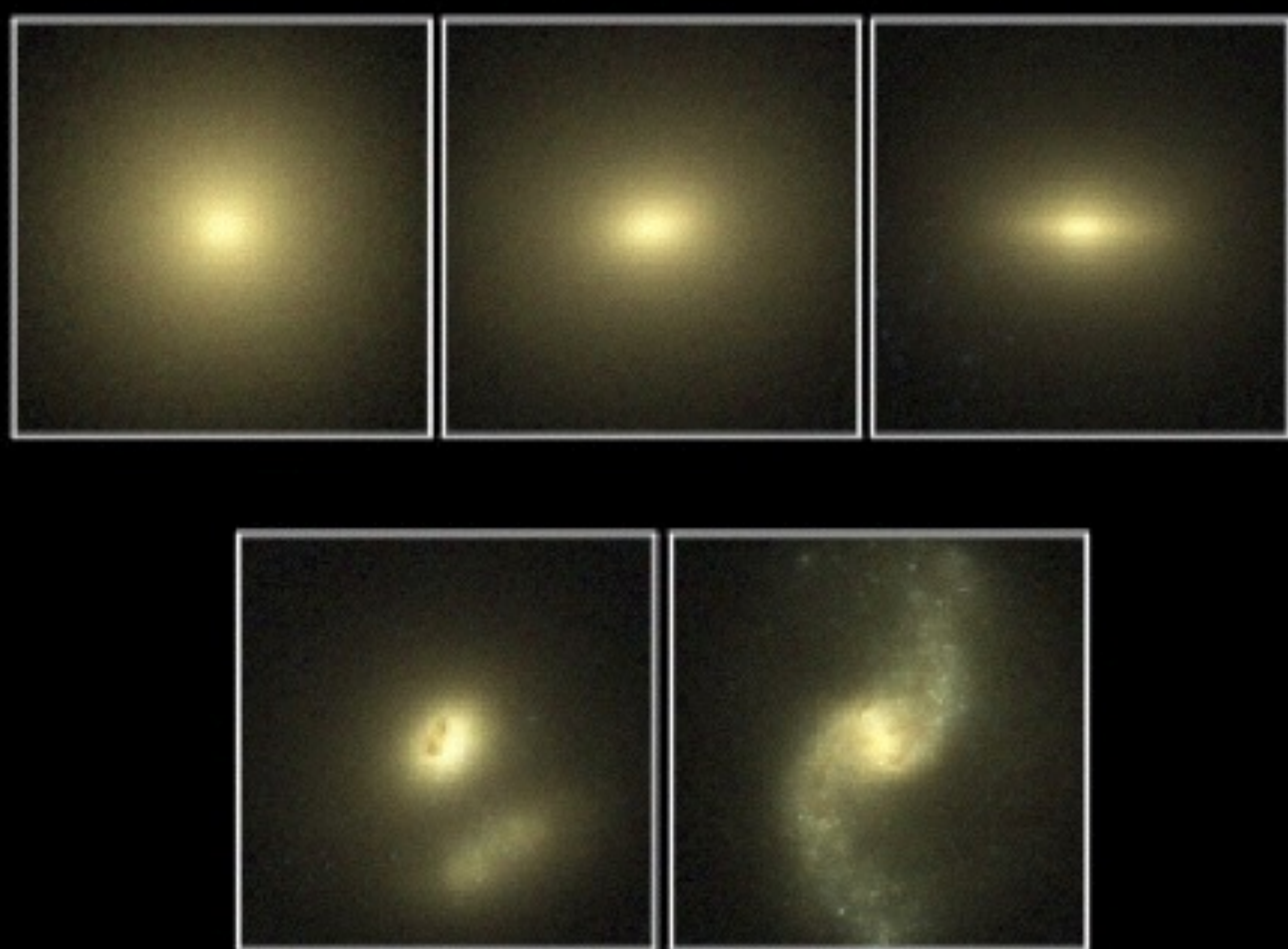
Name	L (comoving Mpc)	N	m_g ($M_\odot$)	m_{dm} ($M_\odot$)	ϵ_{com} (comoving kpc)	ϵ_{prop} (proper kpc)
L025N0376	25	376^3	1.81×10^6	9.70×10^6	2.66	0.70
L025N0752	25	752^3	2.26×10^5	1.21×10^6	1.33	0.35
L050N0752	50	752^3	1.81×10^6	9.70×10^6	2.66	0.70
L100N1504	100	1504^3	1.81×10^6	9.70×10^6	2.66	0.70

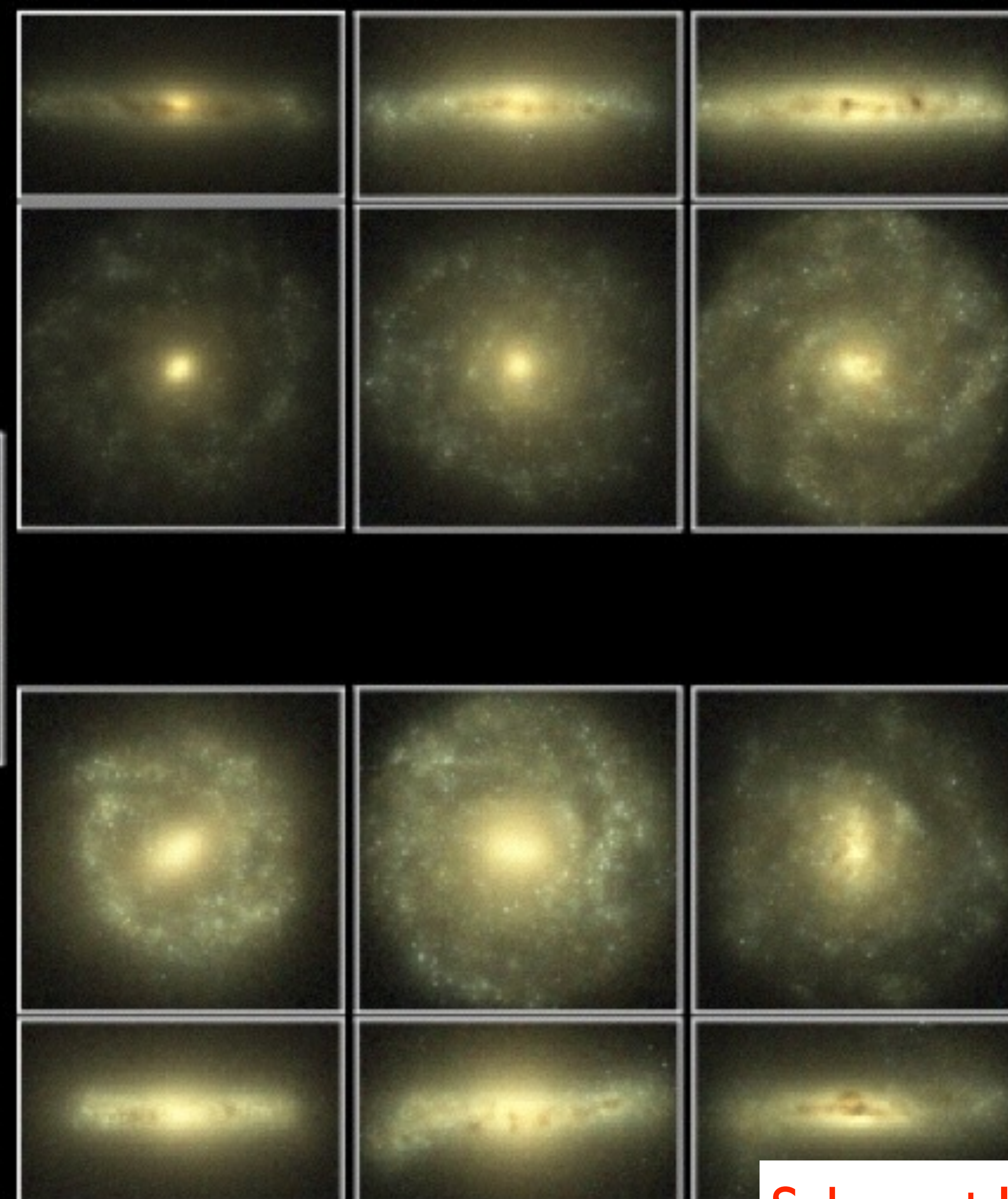
7 M CPU hours

Schaye +15

The Hubble Sequence

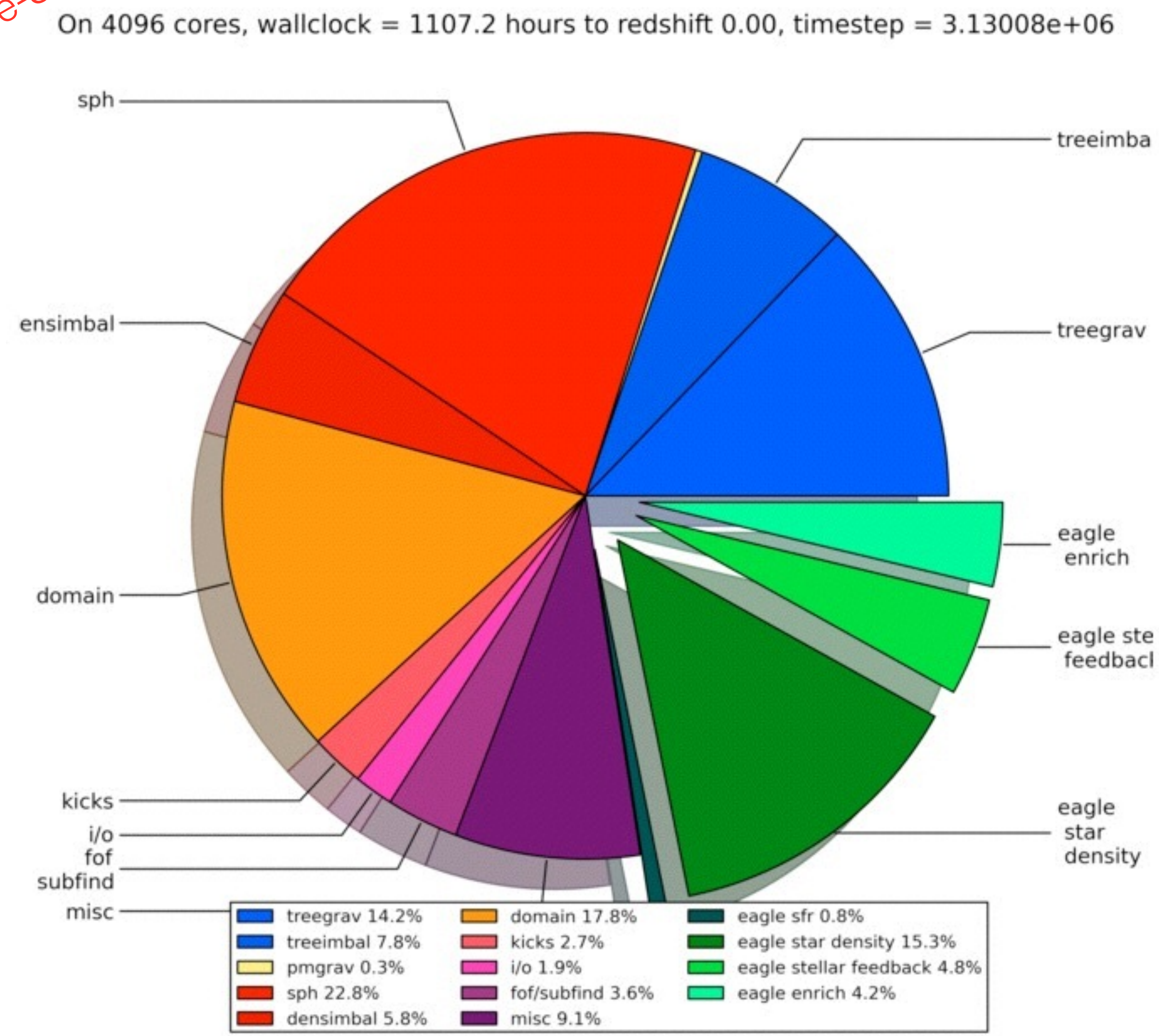






Schaye +15

Gadget-3 Tree-SPH code



SAO/NASA Astrophysics Data System (ADS)

Query Results from the ADS Database

Selected and retrieved **399** abstracts. Total citations: **38878**

#	Bibcode / Authors	Cites / Title	Date	List of Links / Access Control Help
1	2005MNRAS.364.1105S Springel, Volker	2414.000 The cosmological simulation code GADGET-2	12/2005	A F G X R C S O U
2	2005Natur.435..629S Springel, Volker; White, Simon D. M.; Jenkins, Adrian; Frenk, Carlos S.; Yoshida, Naoki; Gao, Liang; Navarro, Julio; Thacker, Robert; Croton, Darren; Helly, John; and 7 coauthors	2199.000 Simulations of the formation, evolution and clustering of galaxies and quasars	06/2005	A E X R C U

Mon. Not. R. Astron. Soc. **364,** 1105–1134 (2005) doi:10.1111/j.1365-2966.2005.09655.x

The cosmological simulation code GADGET-2

Volker Springel★
Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Straße 1, 85740 Garching bei München, Germany

Contents:

1. Lagrangian versus Eulerian dynamics
2. SPH - Smoothed Particle Hydrodynamics
3. Gadget (Tree - SPH)
4. HPC challenges

Continuity equation:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathbf{v})$$

$$\frac{\partial \rho}{\partial t} + \mathbf{v} \cdot \nabla \rho = -\rho \nabla \cdot \mathbf{v}$$

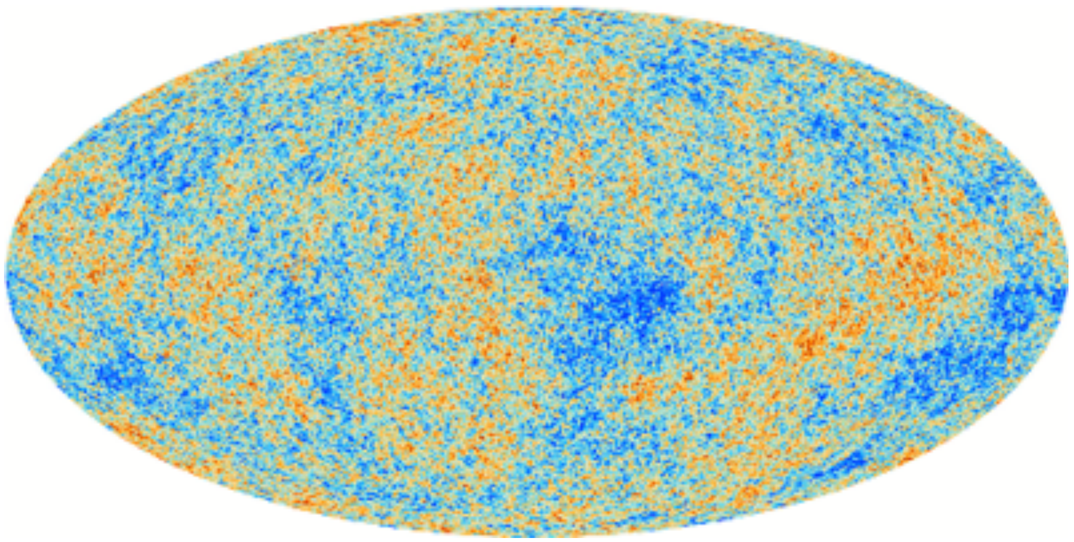
Lagrangian time derivative

$$\frac{\partial}{\partial t} + \mathbf{v} \cdot \nabla \equiv \frac{\mathrm{d}}{dt}$$

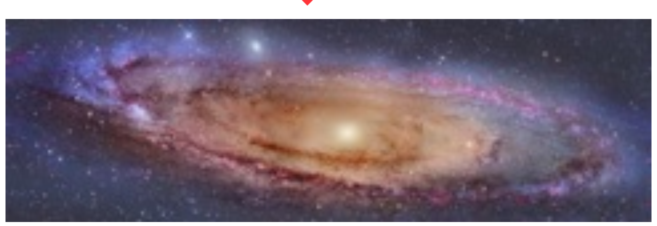
Lagrangian continuity equation:

$$\frac{\mathrm{d}\rho}{\mathrm{d}t} = -\rho \nabla \cdot \mathbf{v}$$

Eulerian versus Lagrangian hydrodynamics

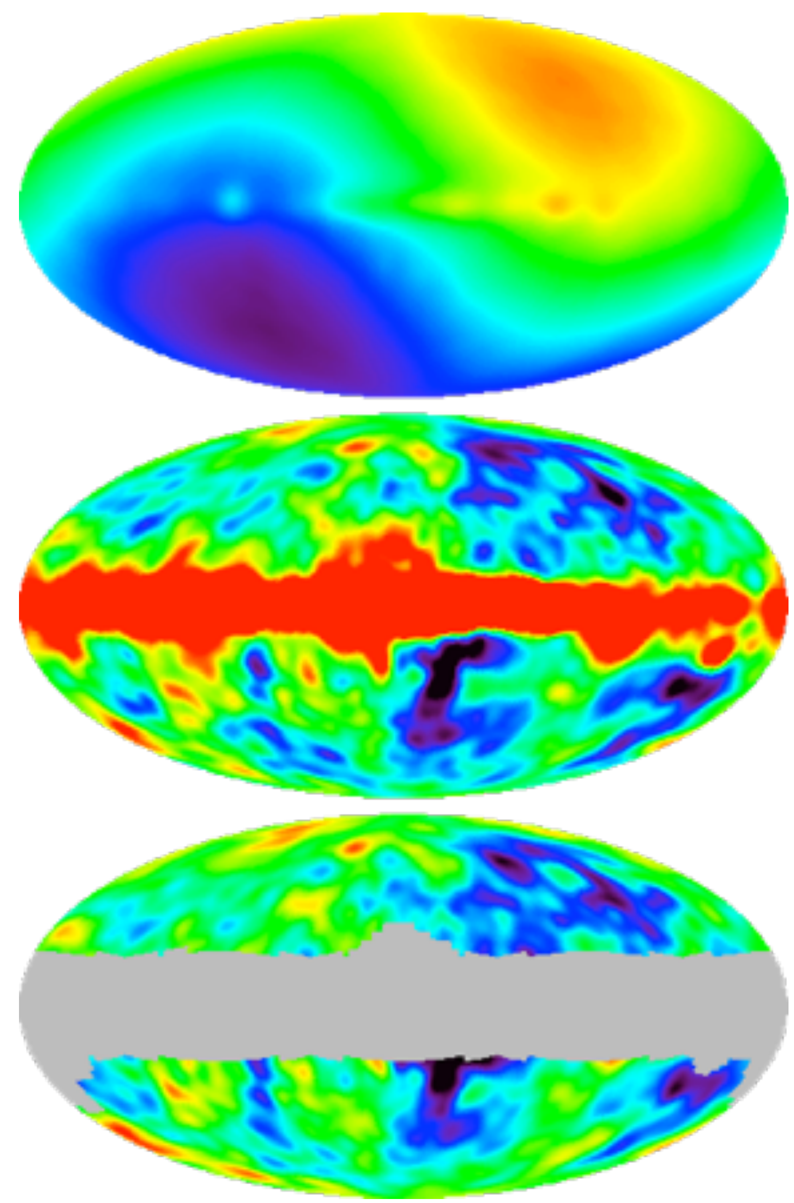


v = 600 km s^{-1}



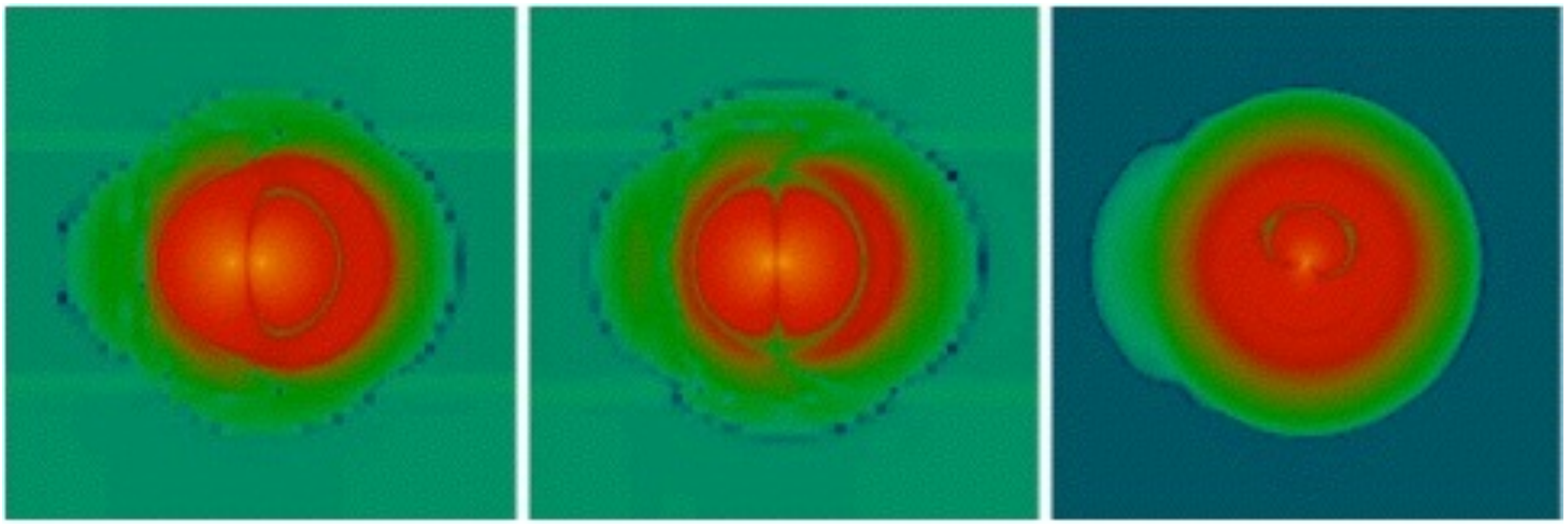
in terms of Mach number:
v/c=600/2=300!

The Universe is isotropic!



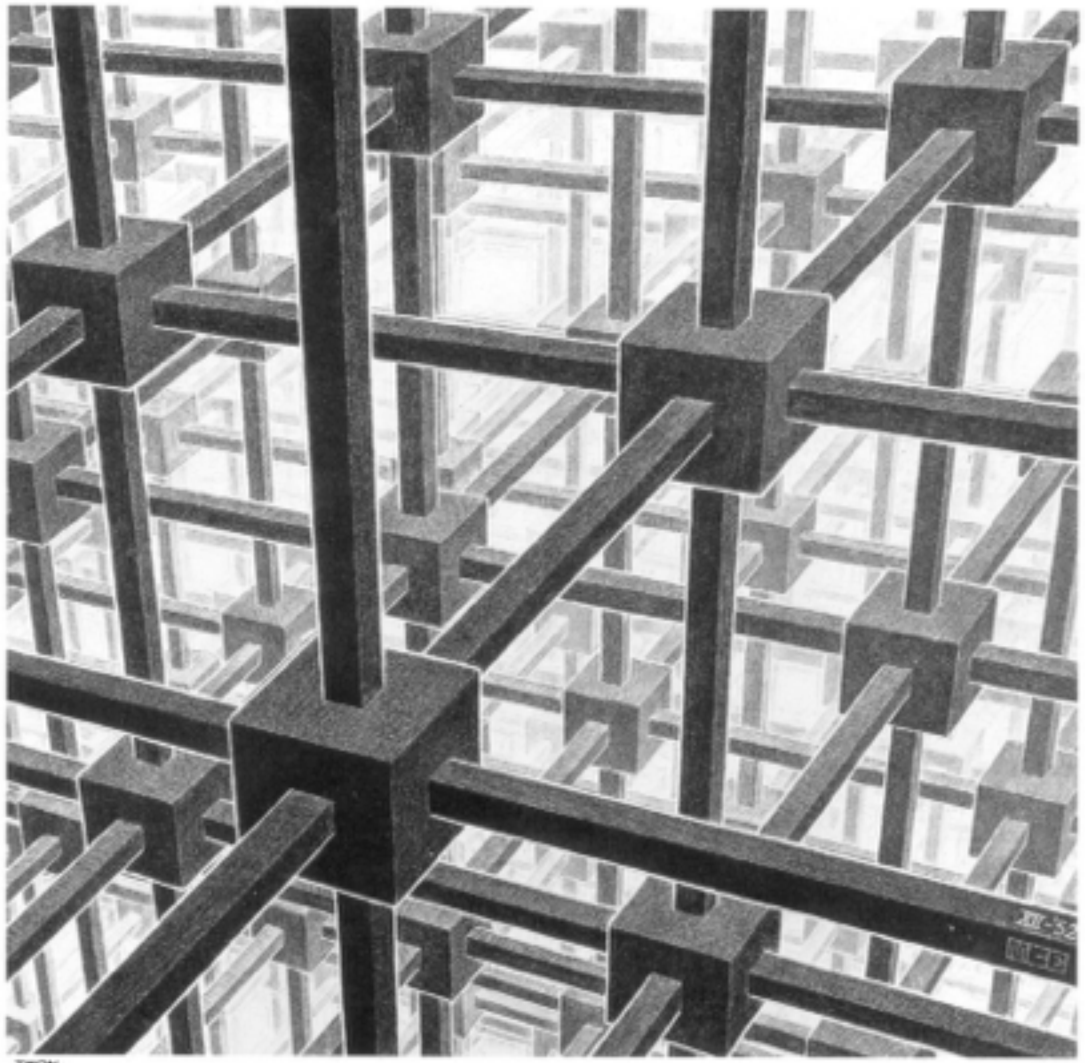
A test suite for quantitative comparison of hydrodynamic codes in astrophysics

Elizabeth J. Tasker,[1★] Riccardo Brunino,[2] Nigel L. Mitchell,[3] Dolf Michielsen,[2] Stephen Hopton,[2] Frazer R. Pearce,[2] Greg L. Bryan[4] and Tom Theuns[3,5]

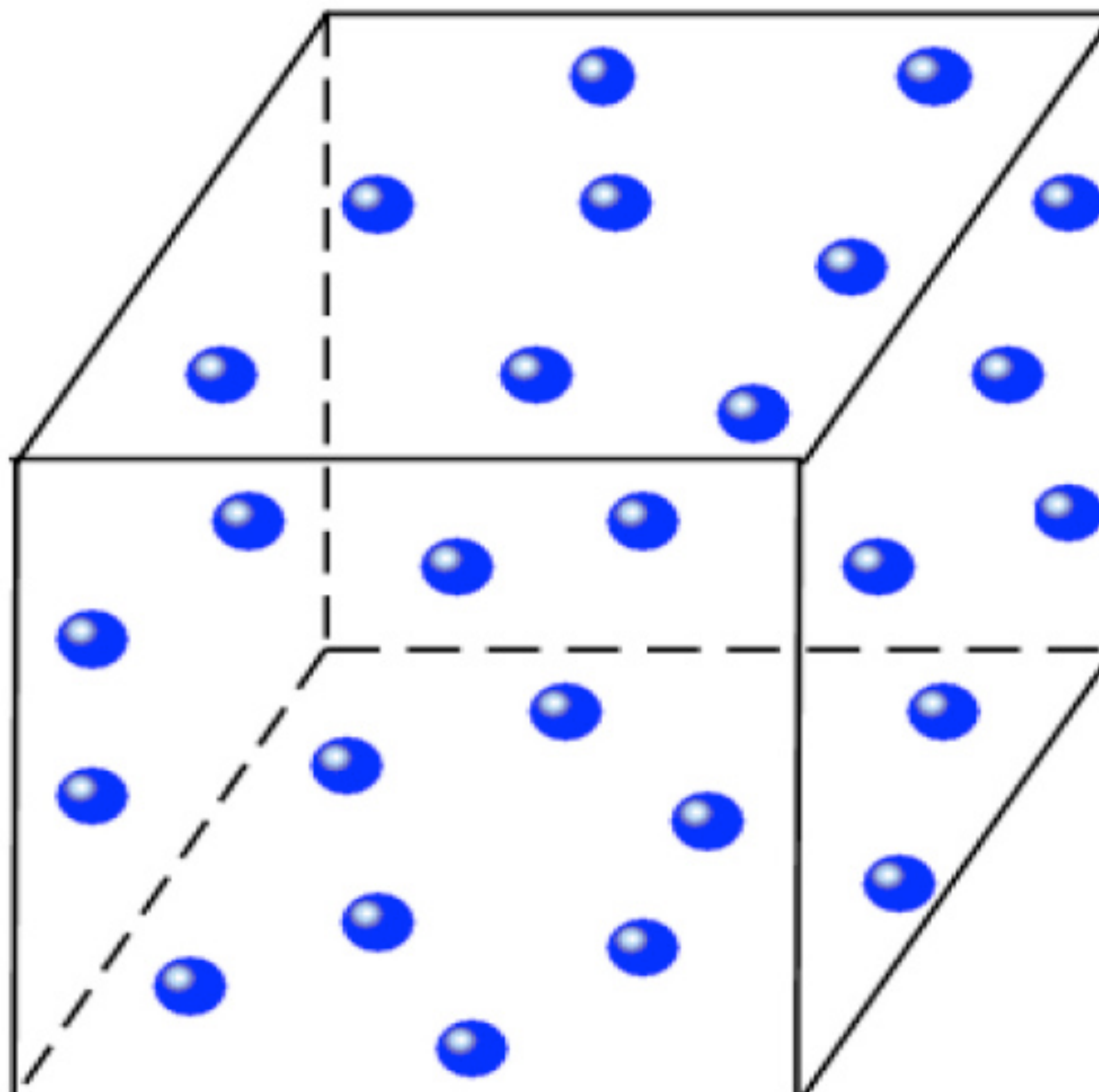


(non)-Galilean invariance of Eulerian codes

Mesh has special directions



Lagrangian distribution is isotropic on average



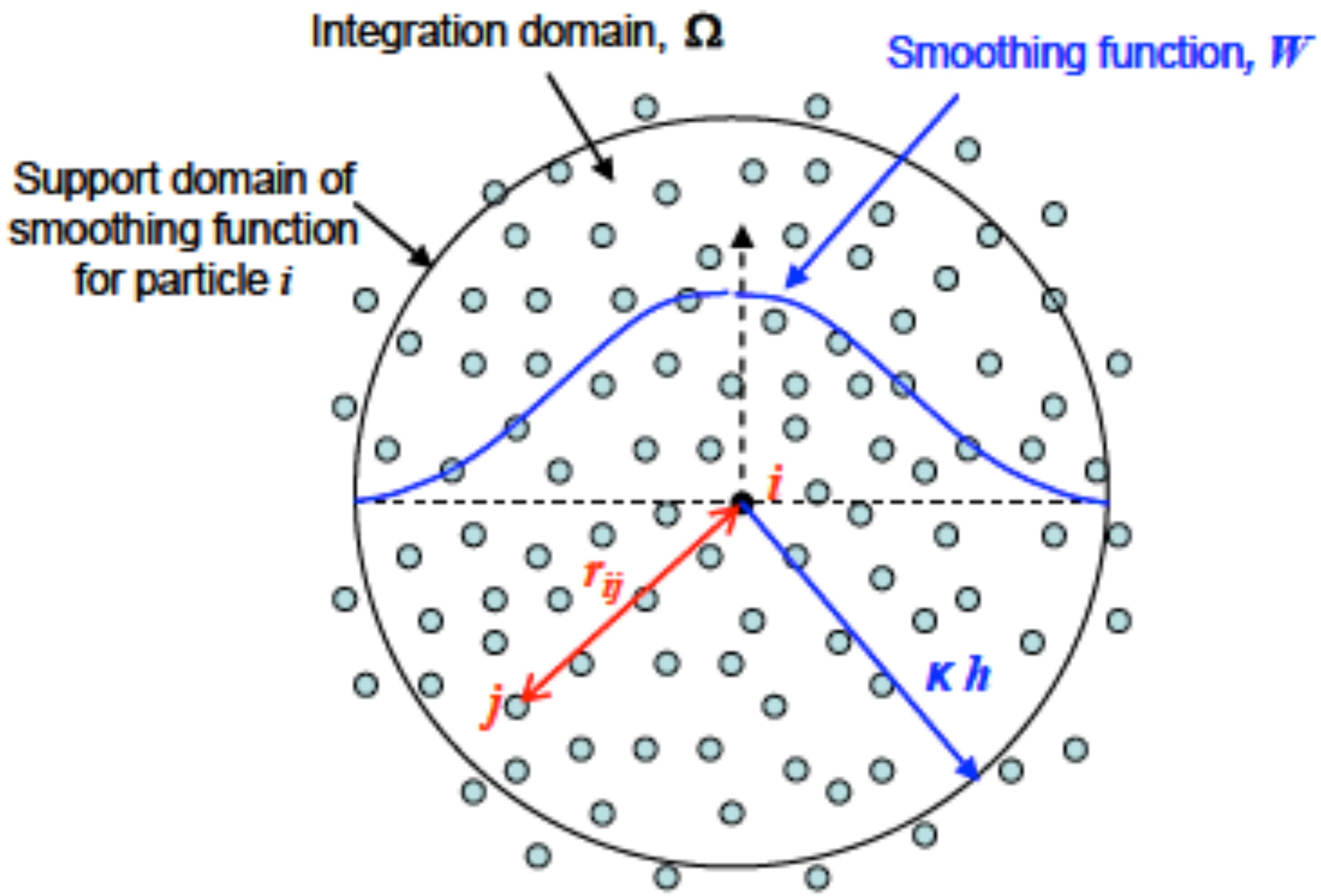
Time-step limitation (Courant step)

$$\Delta t \; c < h$$

sound speed and resolution vary by orders of magnitude

use **individual time-steps**

SPH: Interpolation on a disordered grid



original SPH papers:

- Gingold & Monaghan, MNRAS 181, 1977
- Lucy, AJ 82, 1977

reviews:

- Springel, ARAA 48, 2010
- Price, JComputPhys 231, 2012

Smoothed Particle Hydrodynamics: basics

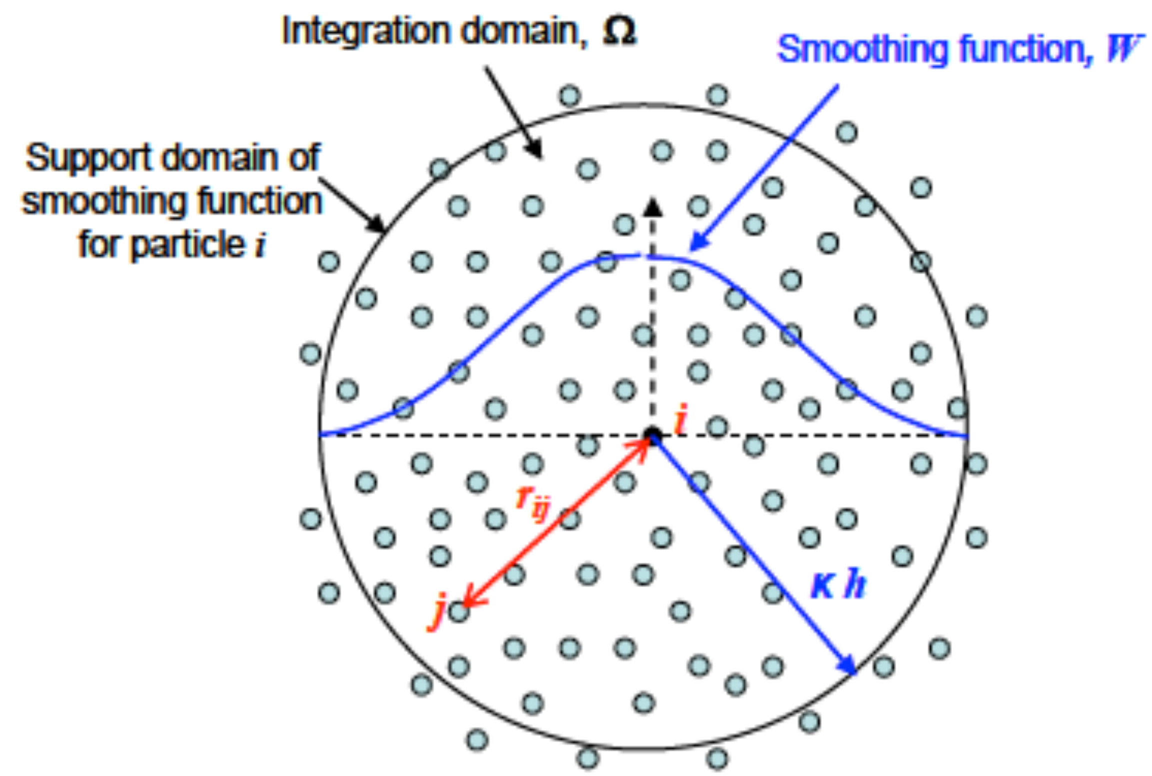
$$\rho(\mathbf{r}_i) = \sum_j m_j W \left(\frac{|\mathbf{r}_i - \mathbf{r}_j|}{h_i} \right)$$

differentiable
compact support

kernel function

particle mass

sum over neighbours
(“close enough”)



algorithmic

- choice of kernel function
- other hydro terms
- (time) integration schema

computational

- neighbour finding
- parallelisation

Other terms

$$\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = -\frac{\nabla p}{\rho}$$ pressure gradient

$$\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N} m_j \left[f_i \frac{P_i}{\rho_i^2} \nabla_i W_{ij}(h_i) + f_j \frac{P_j}{\rho_j^2} \nabla_i W_{ij}(h_j) \right],$$

$$-\nabla \phi$$ gravitational force

$$-\sum_k \frac{\mathbf{G}\, m_k}{r_{ik}^3} \mathbf{r}_{ik}$$

$$\frac{\mathrm{d}S}{\mathrm{d}t} = \mathcal{H} - \mathcal{C}$$ radiative heating and cooling

Equations to compute

LHS | RHS

$$\rho_i = \sum_{j=1}^{N} m_j W(|\mathbf{r}_{ij}|, h_i),$$

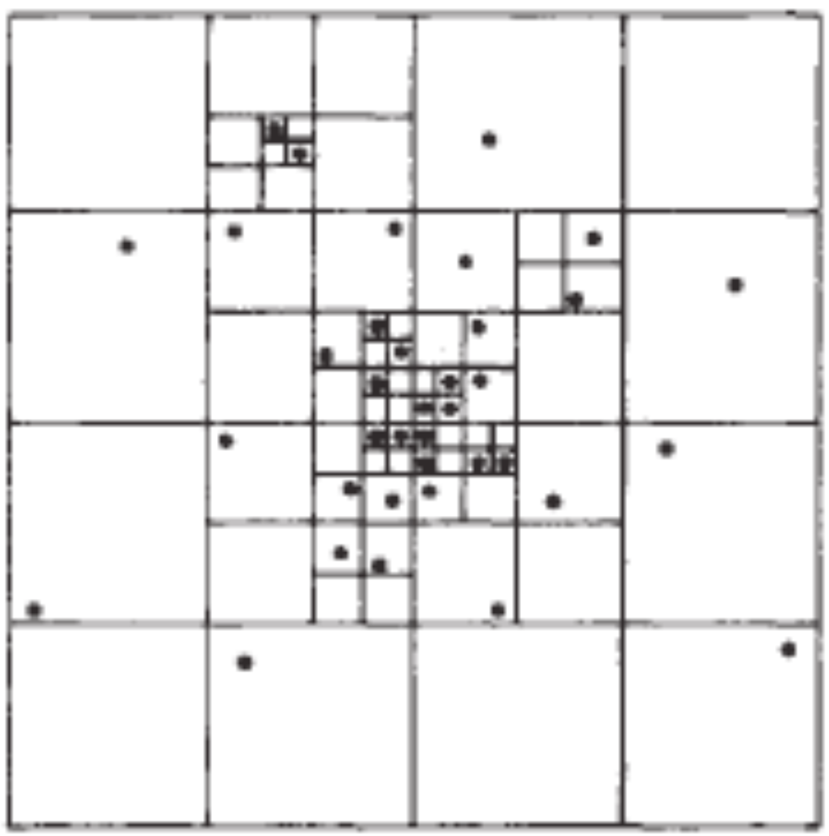
$$\frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N} m_j \left[f_i \frac{P_i}{\rho_i^2} \nabla_i W_{ij}(h_i) + f_j \frac{P_j}{\rho_j^2} \nabla_i W_{ij}(h_j) \right]$$

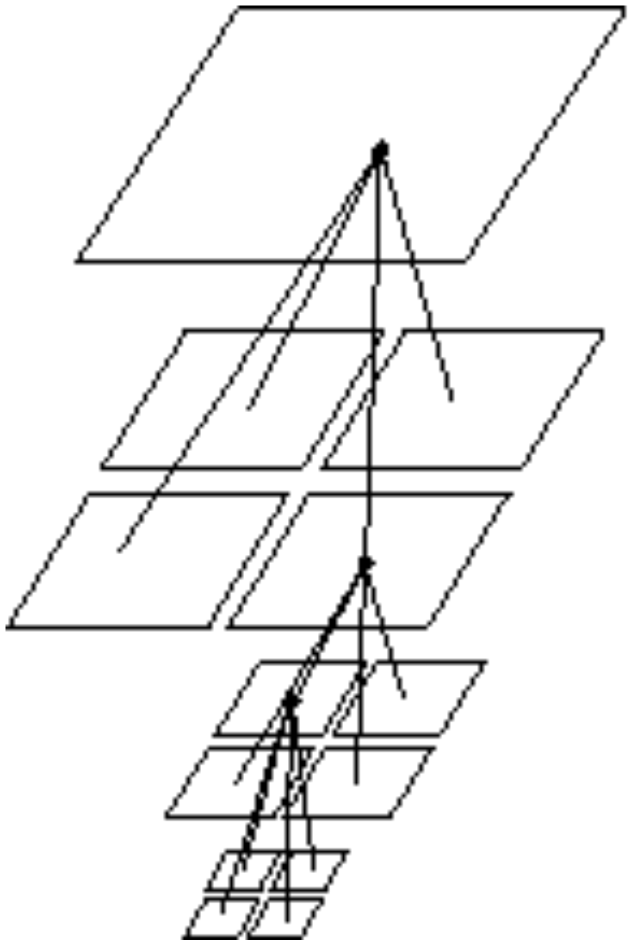
$$-\sum_{j=1}^{N} m_j \Pi_{ij} \nabla_i \overline{W}_{ij},$$

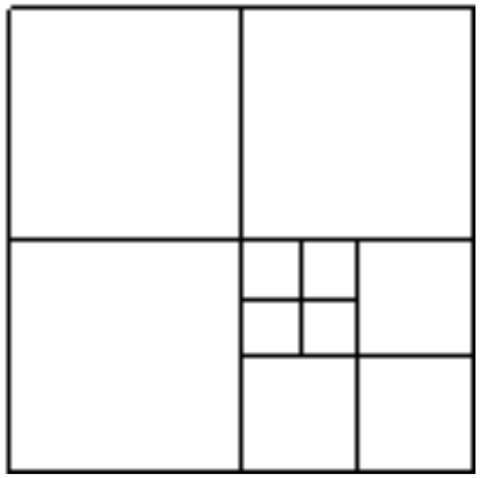
need two neighbour loops

$$\frac{\mathrm{d}A_i}{\mathrm{d}t} = \frac{1}{2} \frac{\gamma - 1}{\rho_i^{\gamma-1}} \sum_{j=1}^{N} m_j \Pi_{ij} \mathbf{v}_{ij} \cdot \nabla_i \overline{W}_{ij}$$

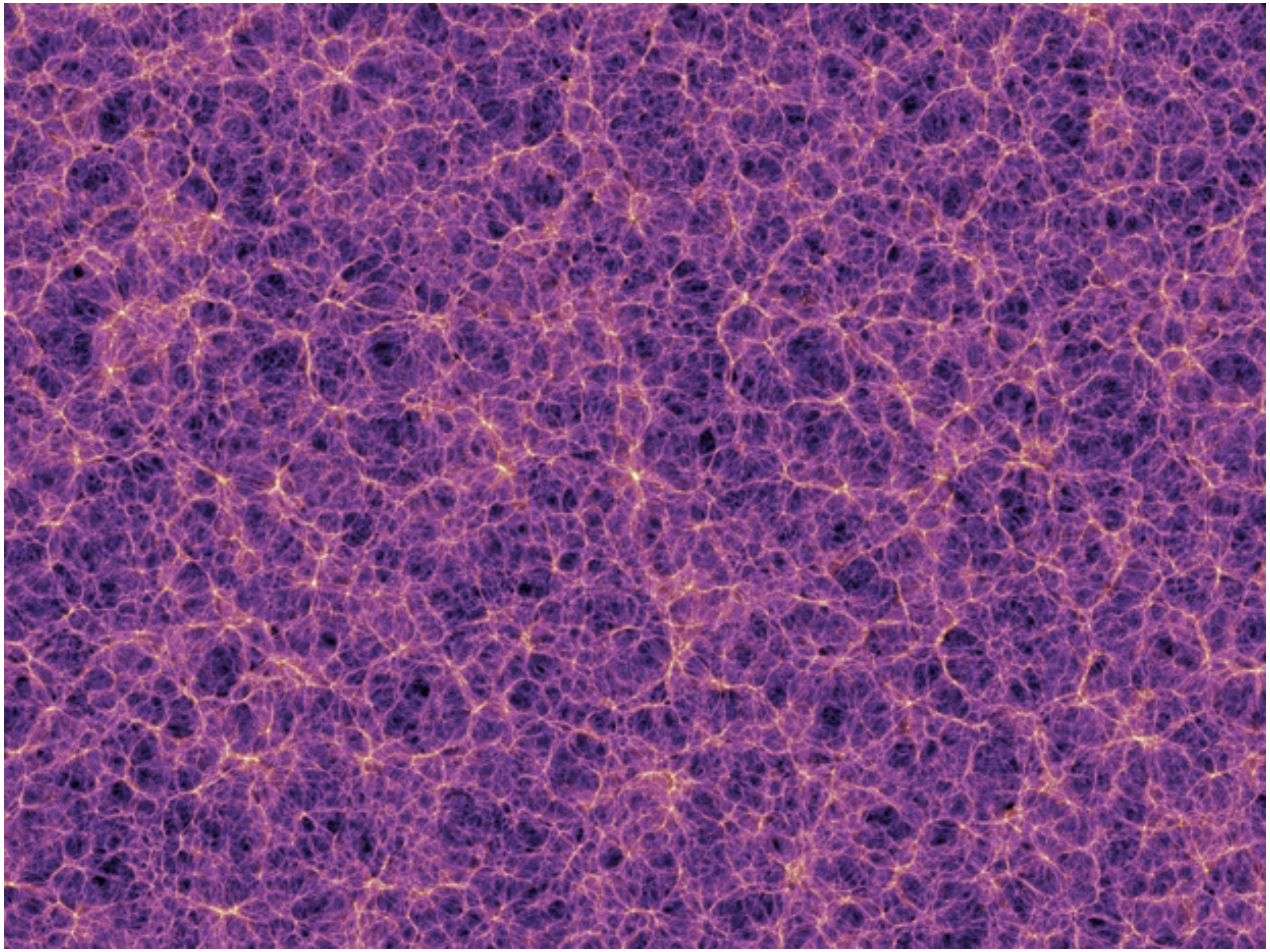
Neighbour finding using a tree



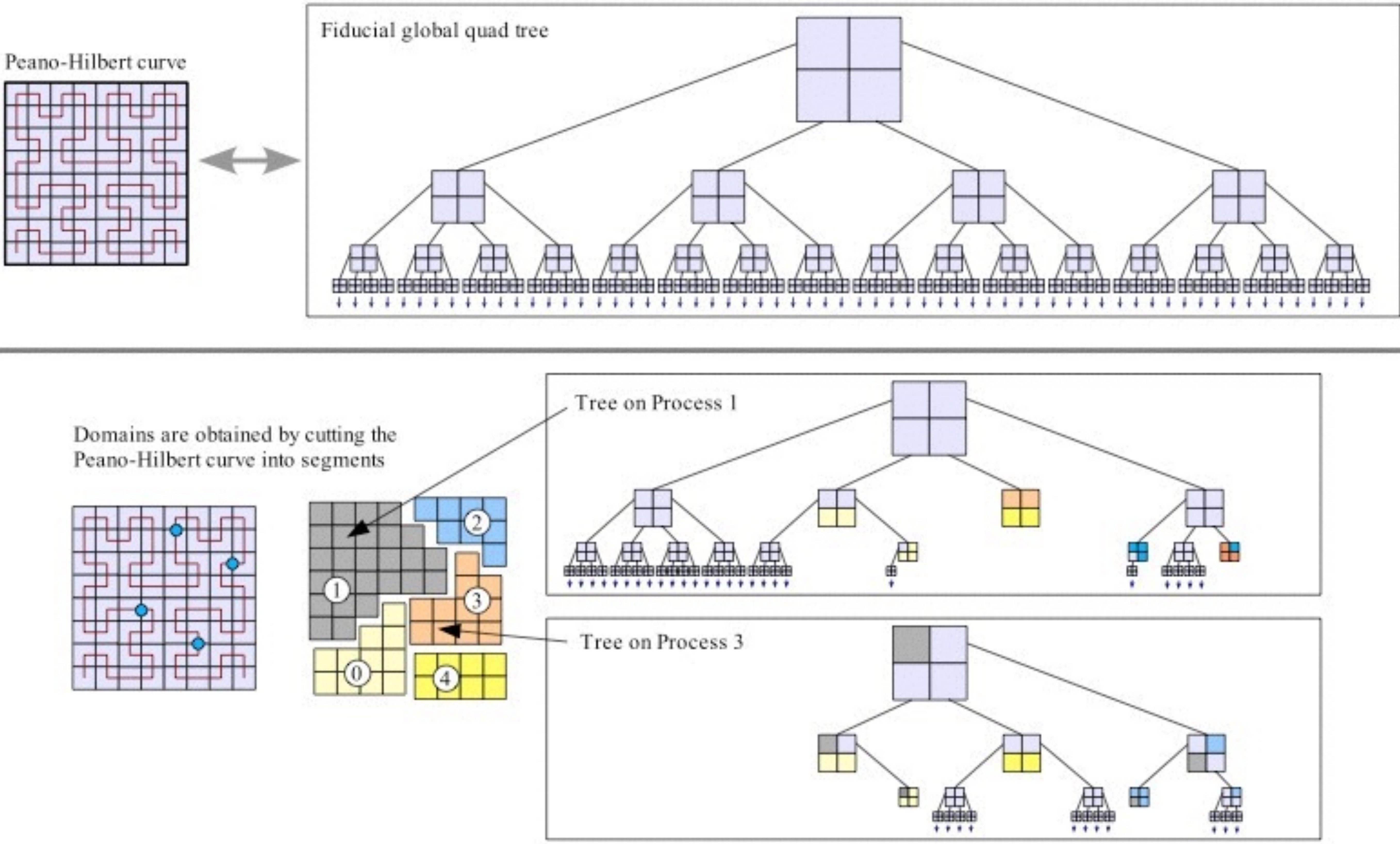




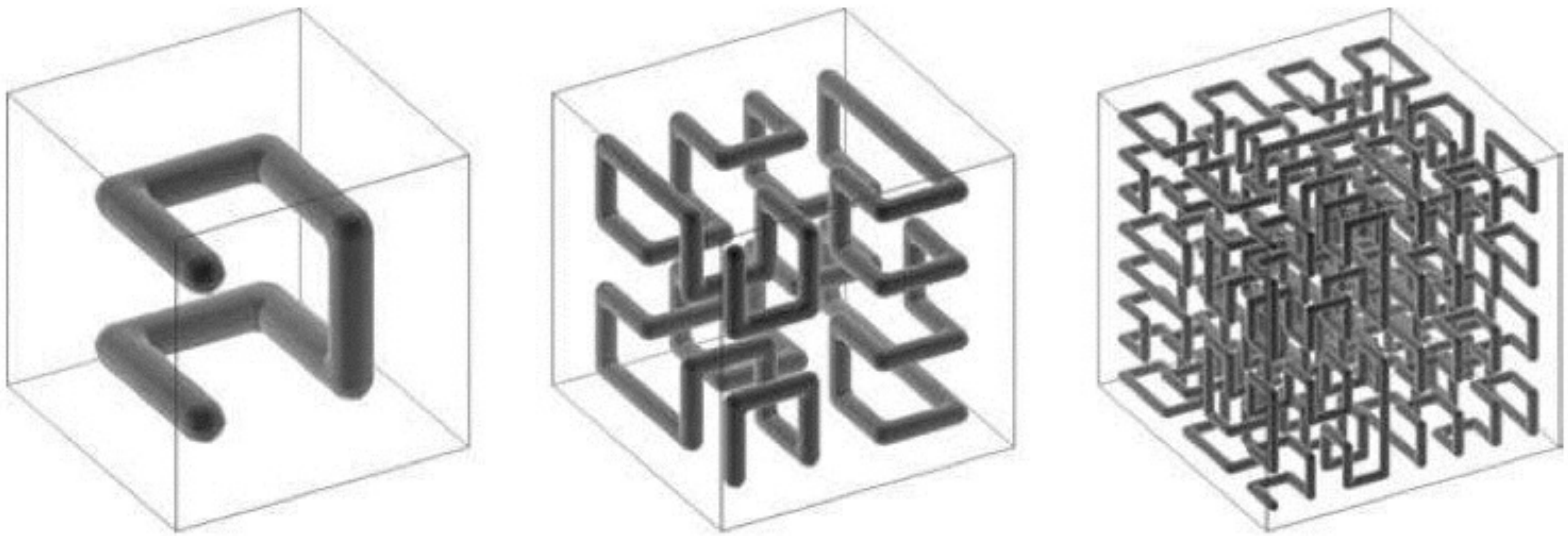
Neighbour finding using a tree



Neighbour finding using a tree



Domain decomposition



time integration

$$\frac{d\mathbf{x}}{dt} = \frac{\mathbf{p}}{m}$$

$$\frac{d\mathbf{p}}{dt} = \mathbf{f}$$

D: Drift

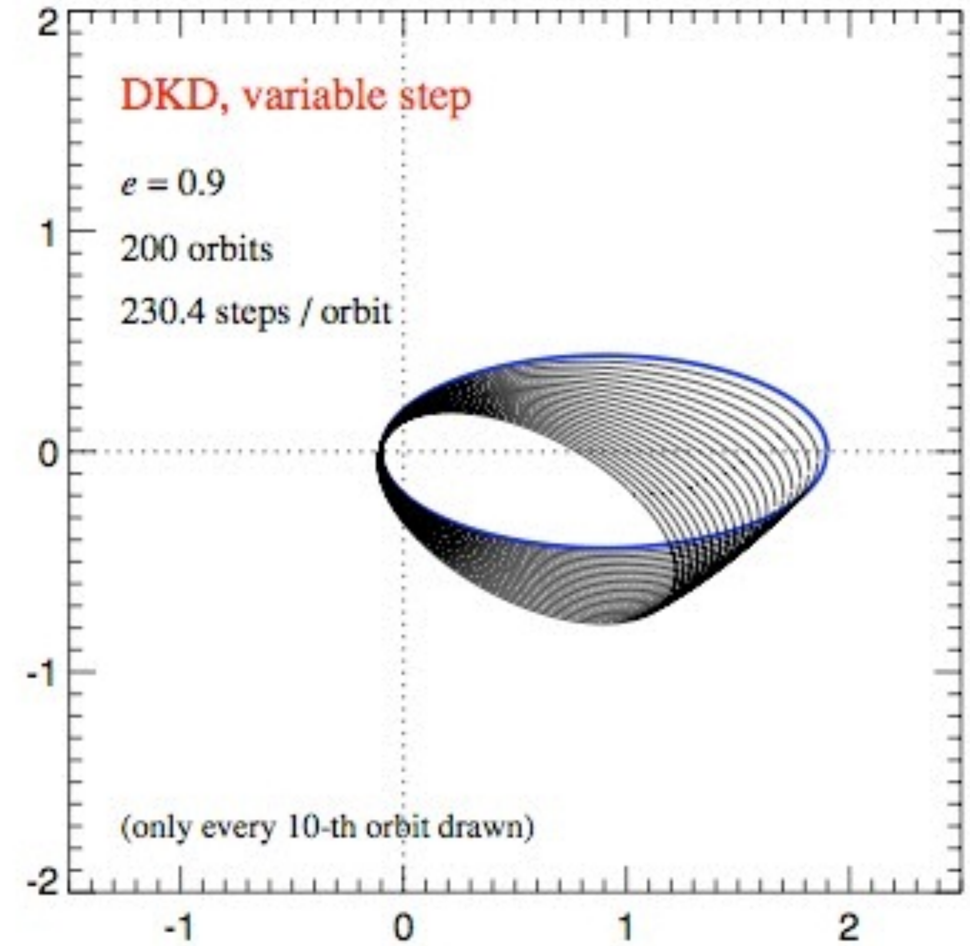
$$D_t(\Delta t) : \begin{cases} \boldsymbol{p}_i & \mapsto \quad \boldsymbol{p}_i \\ \boldsymbol{x}_i & \mapsto \quad \boldsymbol{x}_i + \dfrac{\boldsymbol{p}_i}{m_i} \displaystyle\int_t^{t+\Delta t} \frac{dt}{a^2} \end{cases}$$

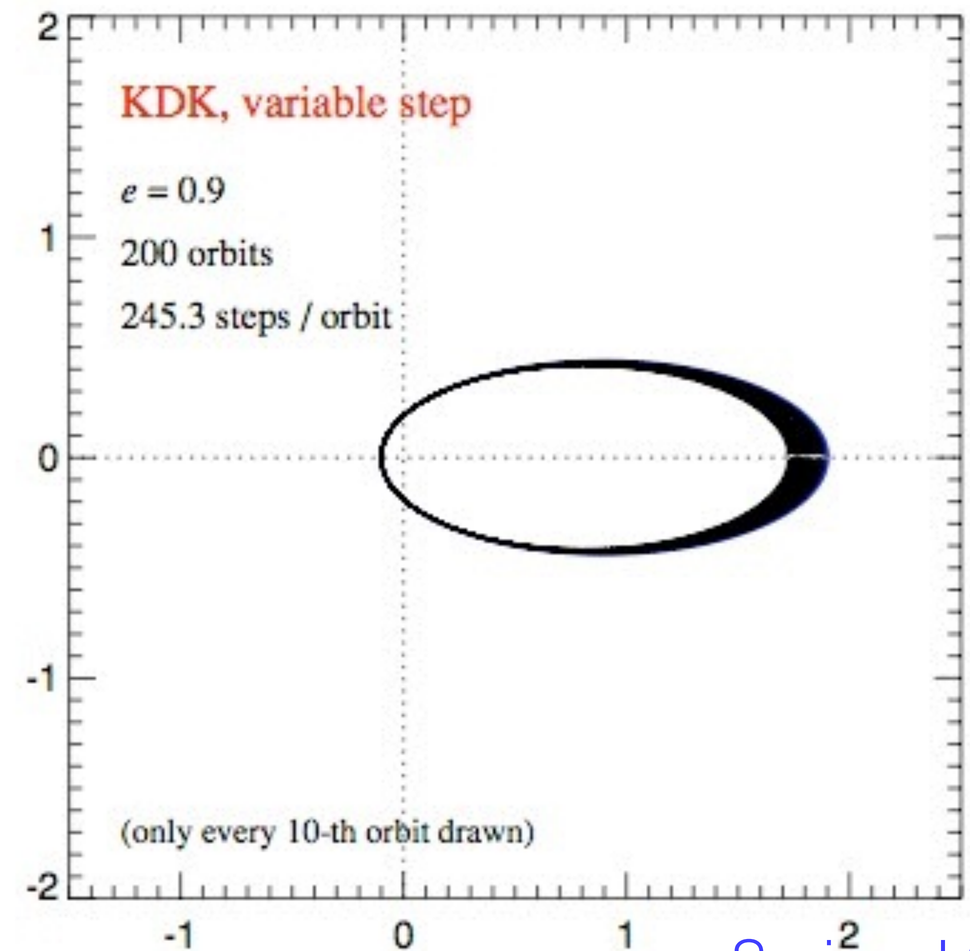
$$K_t(\Delta t) : \begin{cases} \boldsymbol{x}_i & \mapsto \quad \boldsymbol{x}_i \\ \boldsymbol{p}_i & \mapsto \quad \boldsymbol{p}_i + \boldsymbol{f}_i \displaystyle\int_t^{t+\Delta t} \frac{dt}{a} \end{cases}$$

K: Kick

t^n $t^{n+1/2}$ t^{n+1}

x^n x^{n+1}





Springel 05

Implementation in Gadget-2

Mon. Not. R. Astron. Soc. **364,** 1105–1134 (2005) doi:10.1111/j.1365-2966.2005.09655.x

The cosmological simulation code GADGET-2

Volker Springel★

Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Straße 1, 85740 Garching bei München, Germany

Equations to compute

LHS RHS

$$\rho_i = \sum_{j=1}^{N} m_j W(|\boldsymbol{r}_{ij}|, h_i),$$

$$\frac{\mathrm{d}\boldsymbol{v}_i}{\mathrm{d}t} = -\sum_{j=1}^{N} m_j \left[f_i \frac{P_i}{\rho_i^2} \nabla_i W_{ij}(h_i) + f_j \frac{P_j}{\rho_j^2} \nabla_i W_{ij}(h_j) \right]$$

$$-\sum_{j=1}^{N} m_j \Pi_{ij} \nabla_i \overline{W}_{ij},$$

need two neighbour loops

$$\frac{\mathrm{d}A_i}{\mathrm{d}t} = \frac{1}{2} \frac{\gamma - 1}{\rho_i^{\gamma-1}} \sum_{j=1}^{N} m_j \Pi_{ij} \boldsymbol{v}_{ij} \cdot \nabla_i \overline{W}_{ij}$$

$$\sum_{j=1}^{N} m_j W(|\boldsymbol{r}_{ij}|, h_i),$$

density.c

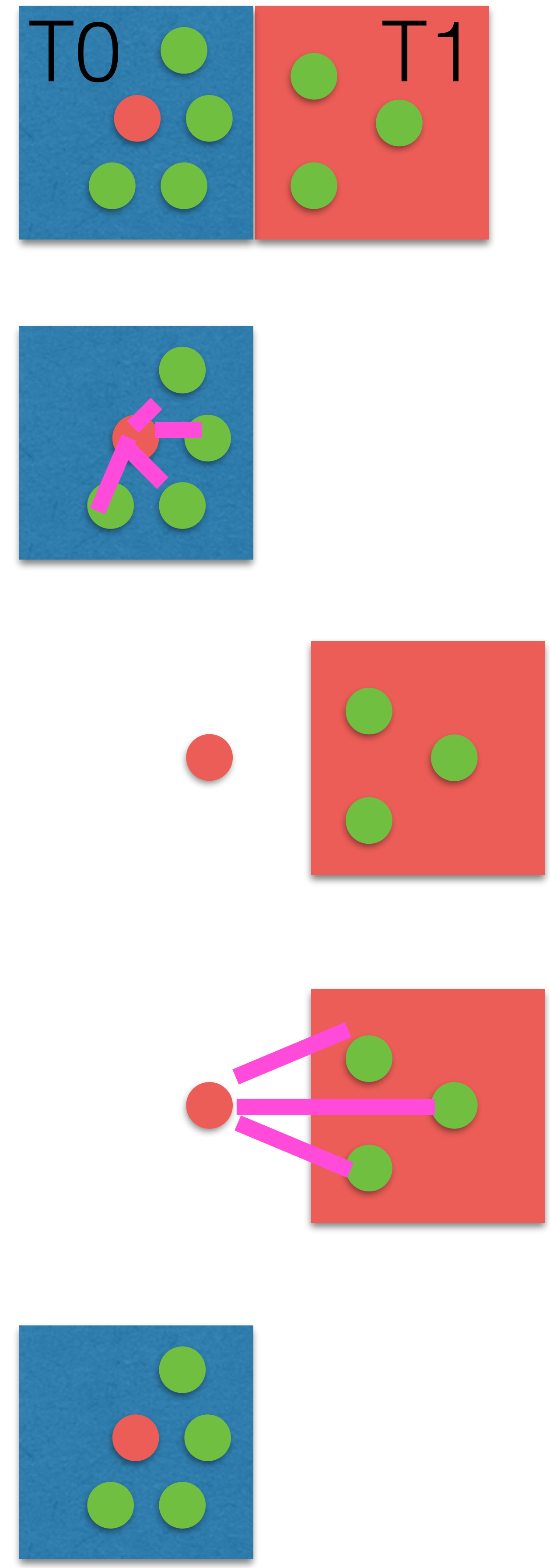
local contribution + export list

export particles

off task contribution

import particles

add imported contributions



local contribution + export list

```
for(j = 0; j < NTask; j++)
  Exportflag[j] = 0;

density_evaluate(i, 0);
```

export particles

```
MPI_Allgather(nsend_local, NTask, MPI_INT, nsend, NTask, MPI_INT, MPI_COMM_WORLD);
```

```
/* get the particles */
MPI_Sendrecv(&DensDataIn[noffset[recvTask]],
        nsend_local[recvTask] * sizeof(struct densdata_in), MPI_BYTE,
```

off-task contribution

```
density_evaluate(j, 1);
```

re-import particles

```
/* send the results */
MPI_Sendrecv(&DensDataResult[nbuffer[ThisTask]],
        nsend[recvTask * NTask + ThisTask] * sizeof(struct densdata_out),
```

add imported contributions

```
SphP[place].Density += DensDataPartialResult[source].Rho;
```

local contribution + export list

```
for(j = 0; j < NTask; j++)
  Exportflag[j] = 0;

density_evaluate(i, 0);
```

MPI-synchronisation

export particles

```
MPI_Allgather(nsend_local, NTask, MPI_INT, nsend, NTask, MPI_INT, MPI_COMM_WORLD);
/* get the particles */
MPI_Sendrecv(&DensDataIn[noffset[recvTask]],
        nsend_local[recvTask] * sizeof(struct densdata_in), MPI_BYTE,
```

off-task contribution

```
density_evaluate(j, 1);
```

re-import particles

```
/* send the results */
MPI_Sendrecv(&DensDataResult[nbuffer[ThisTask]],
        nsend[recvTask * NTask + ThisTask] * sizeof(struct densdata_out),
```

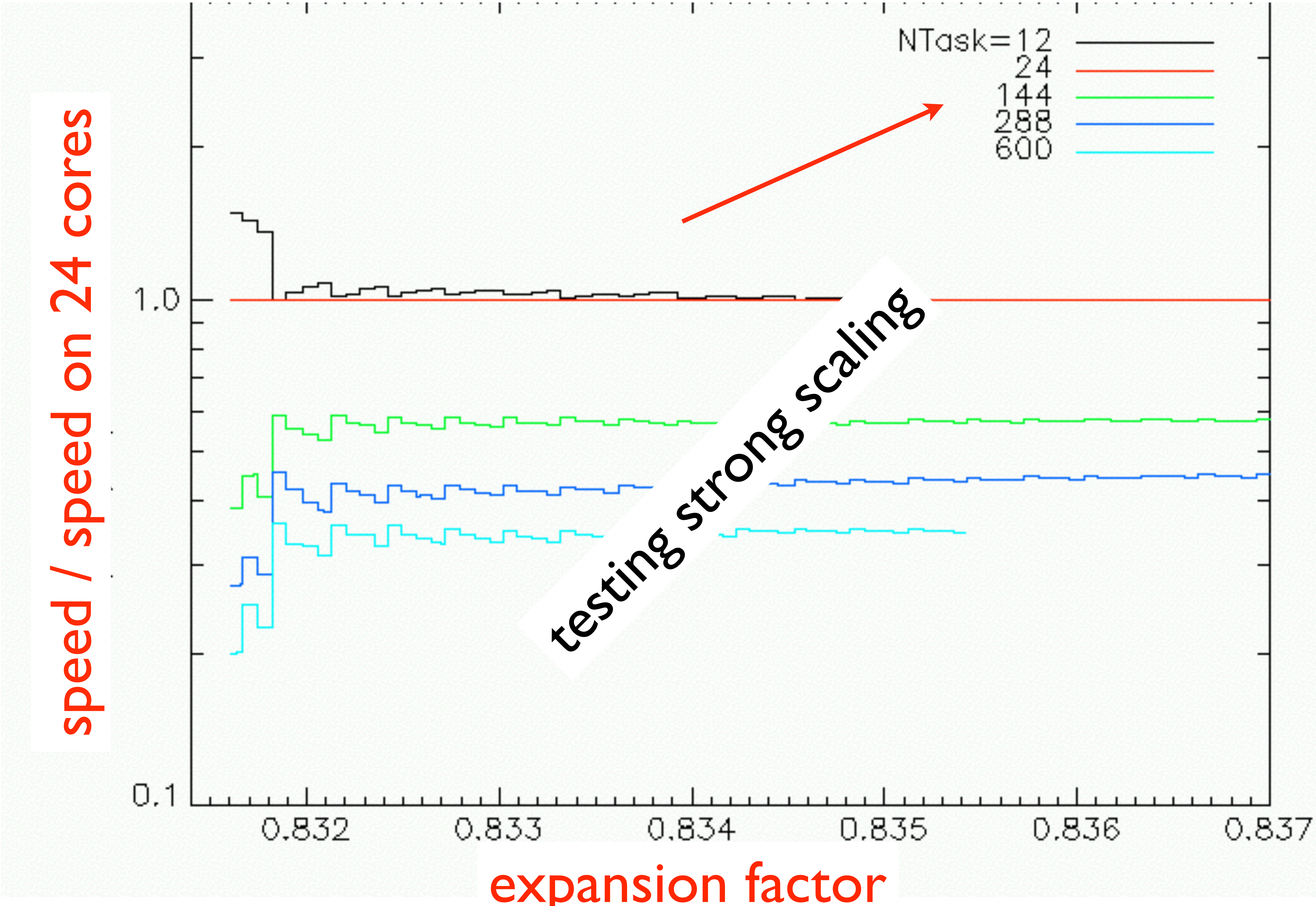
add imported contributions

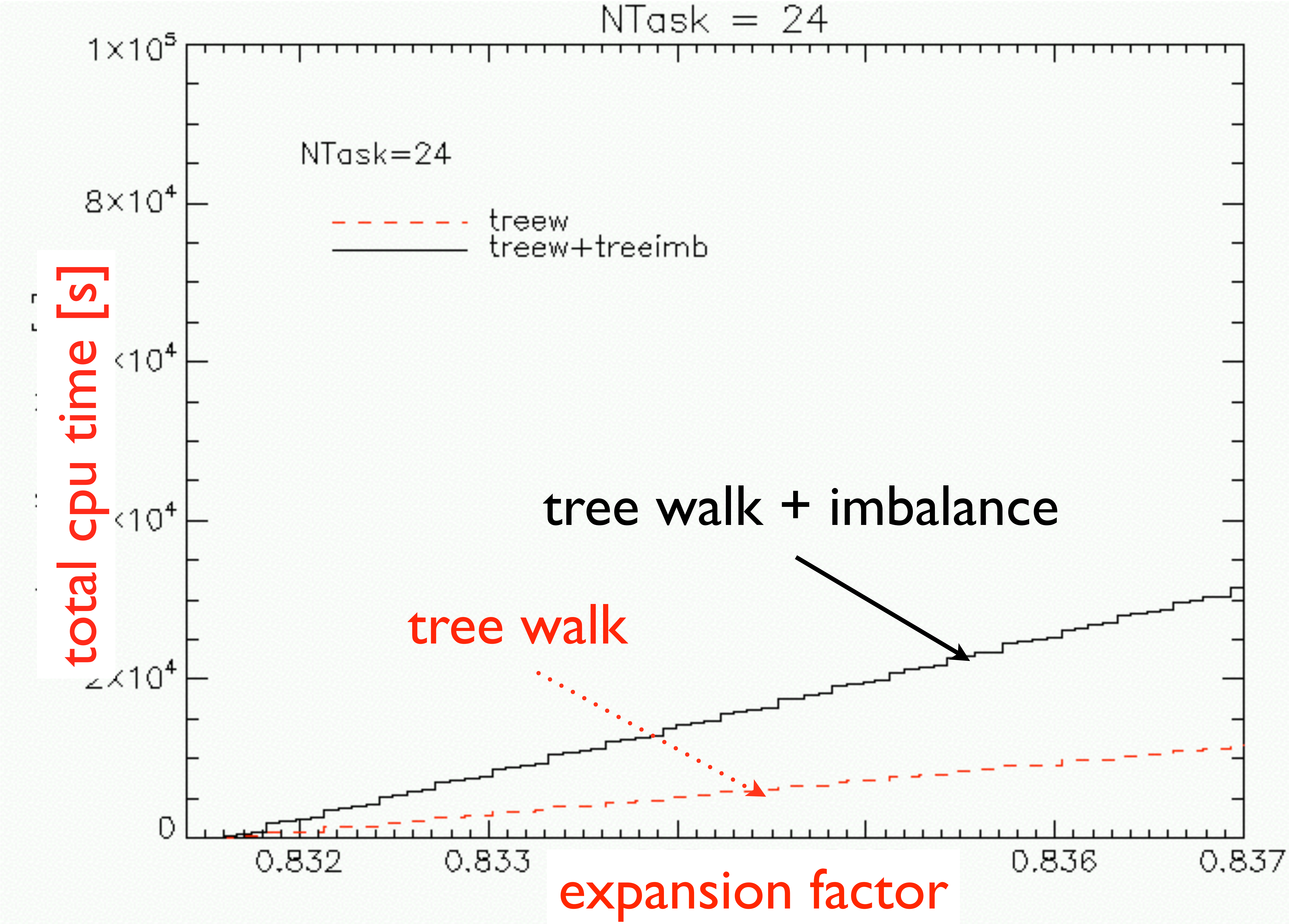
```
SphP[place].Density += DensDataPartialResult[source].Rho;
```

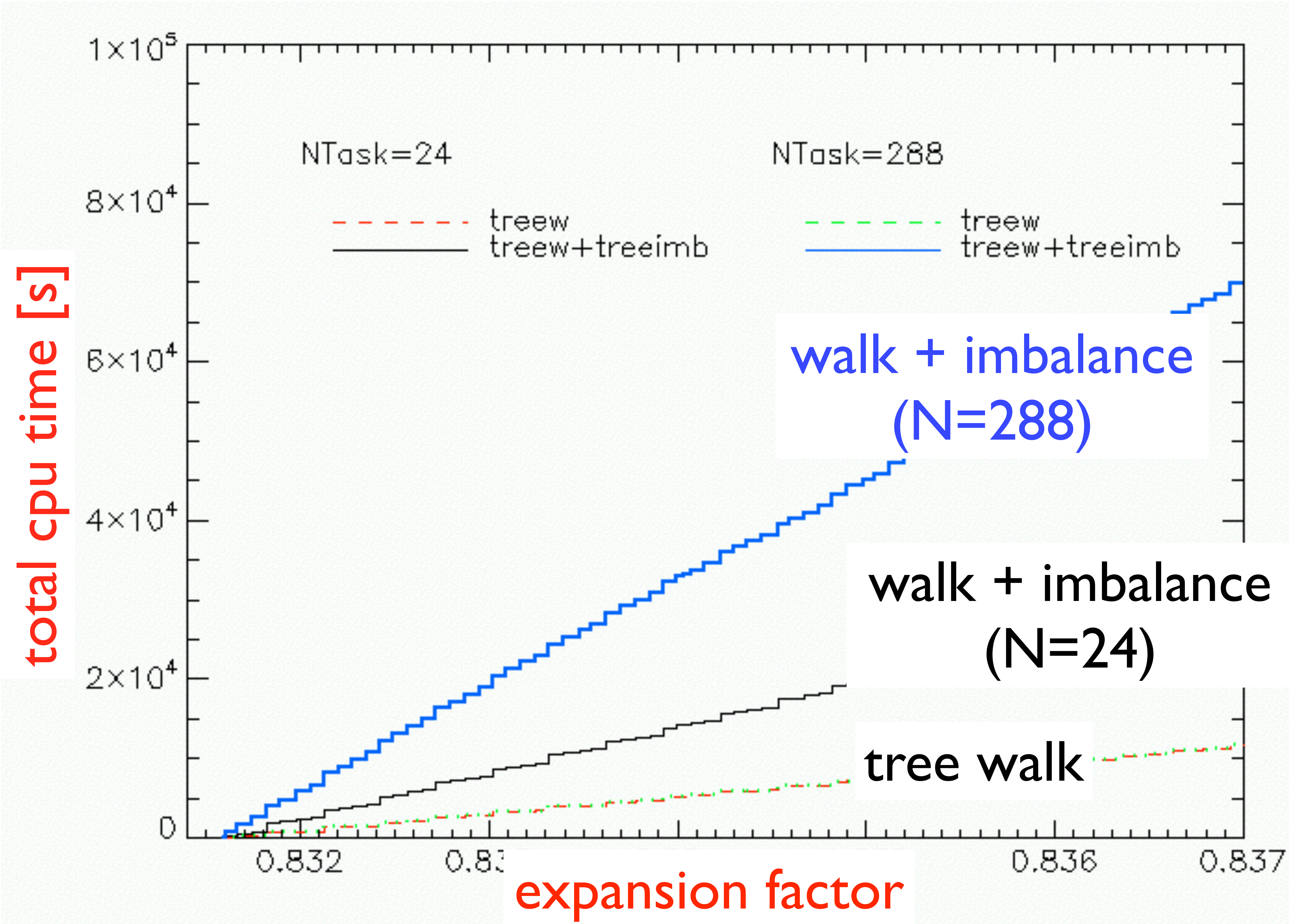
Gravity scaling:

Aquarius (Springel+08)

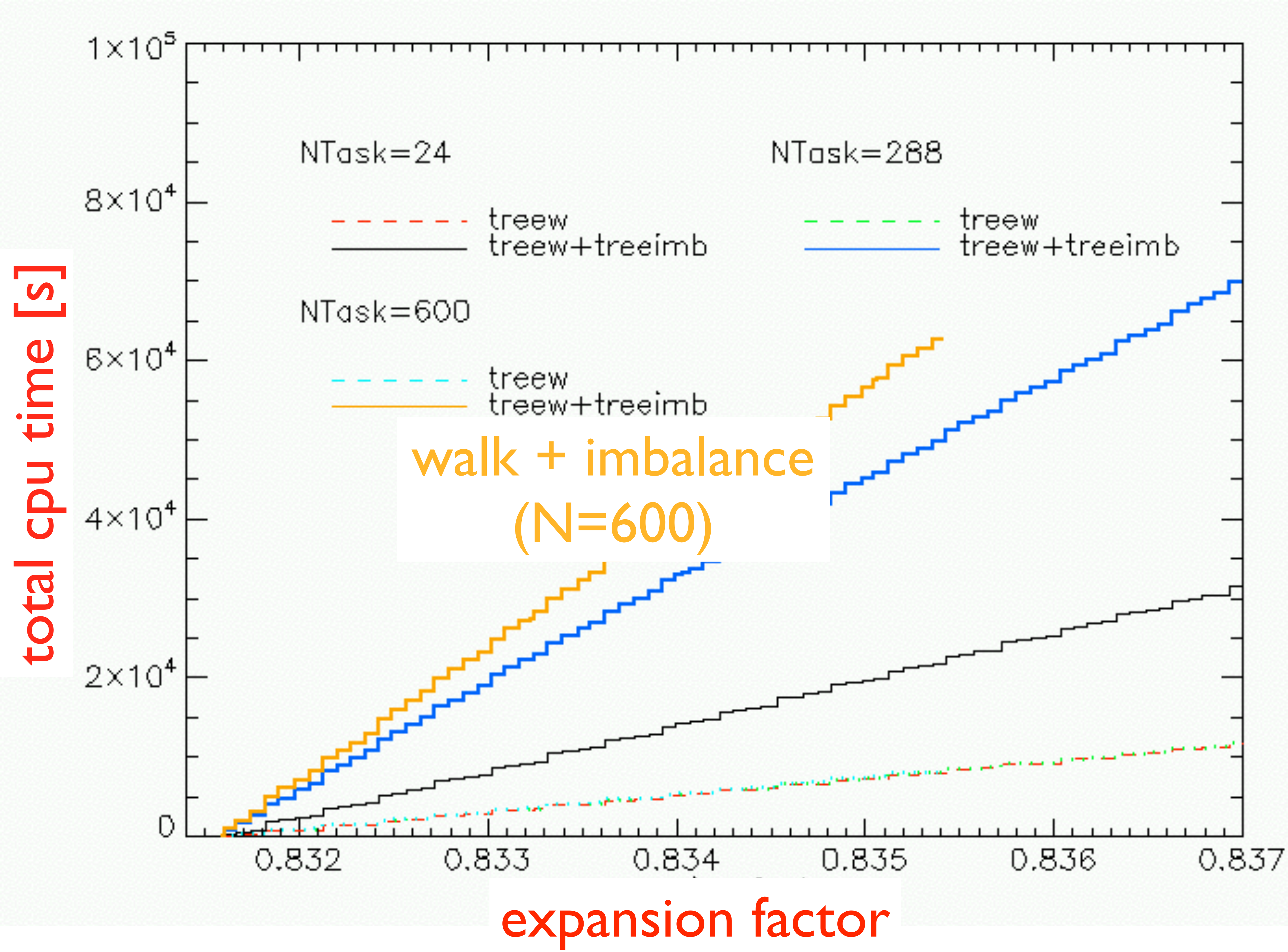
Timings of Gadget-2 on Aquarius Aq-A-4 ($2\ 10^7$ particles, DM only)



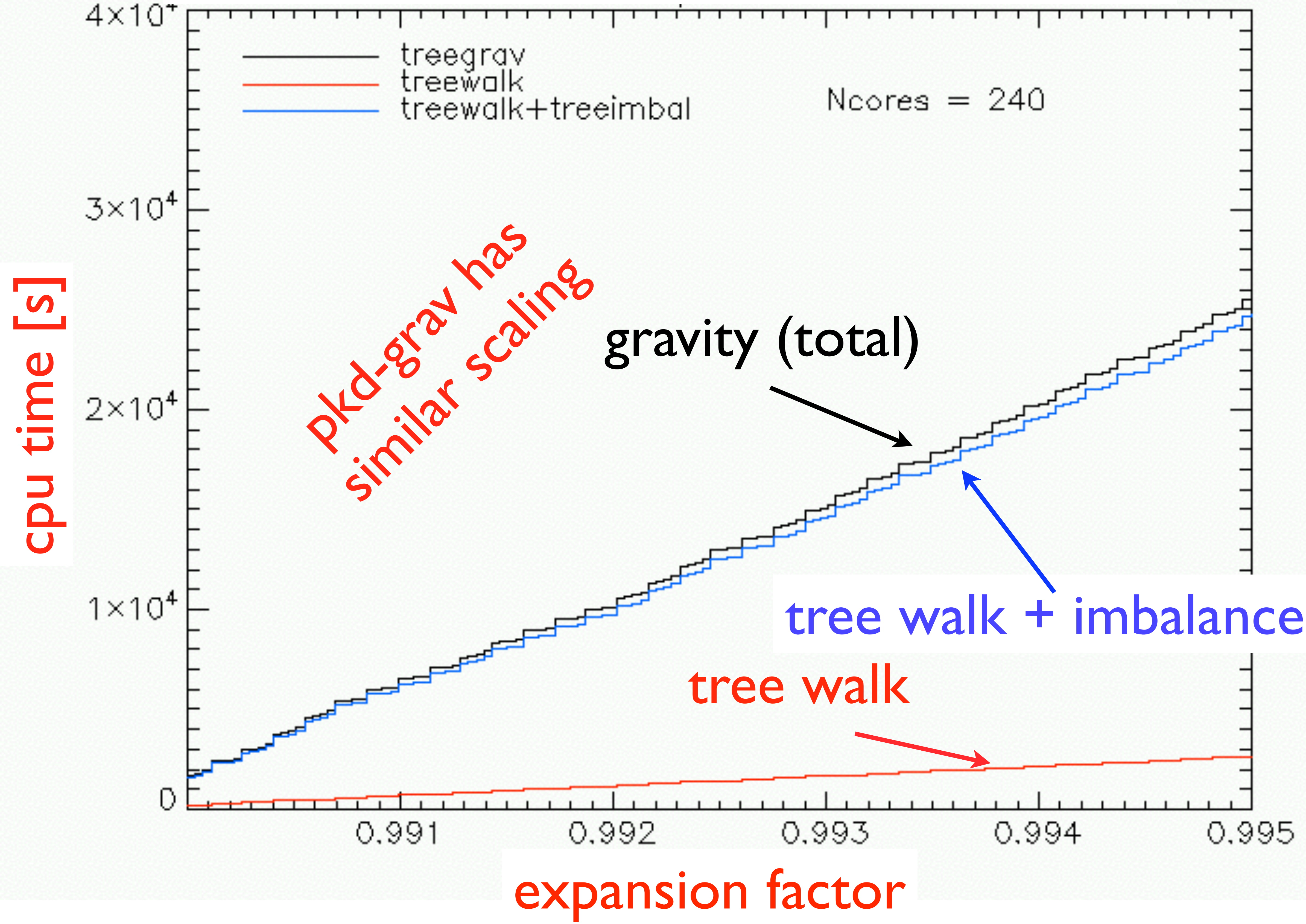

NTask = 24
NTask=24
treew
treew+treeimb
tree walk + imbalance
tree walk
total cpu time [s]
expansion factor
1×10⁵
8×10⁴
0
0.832
0.833
0.836
0.837



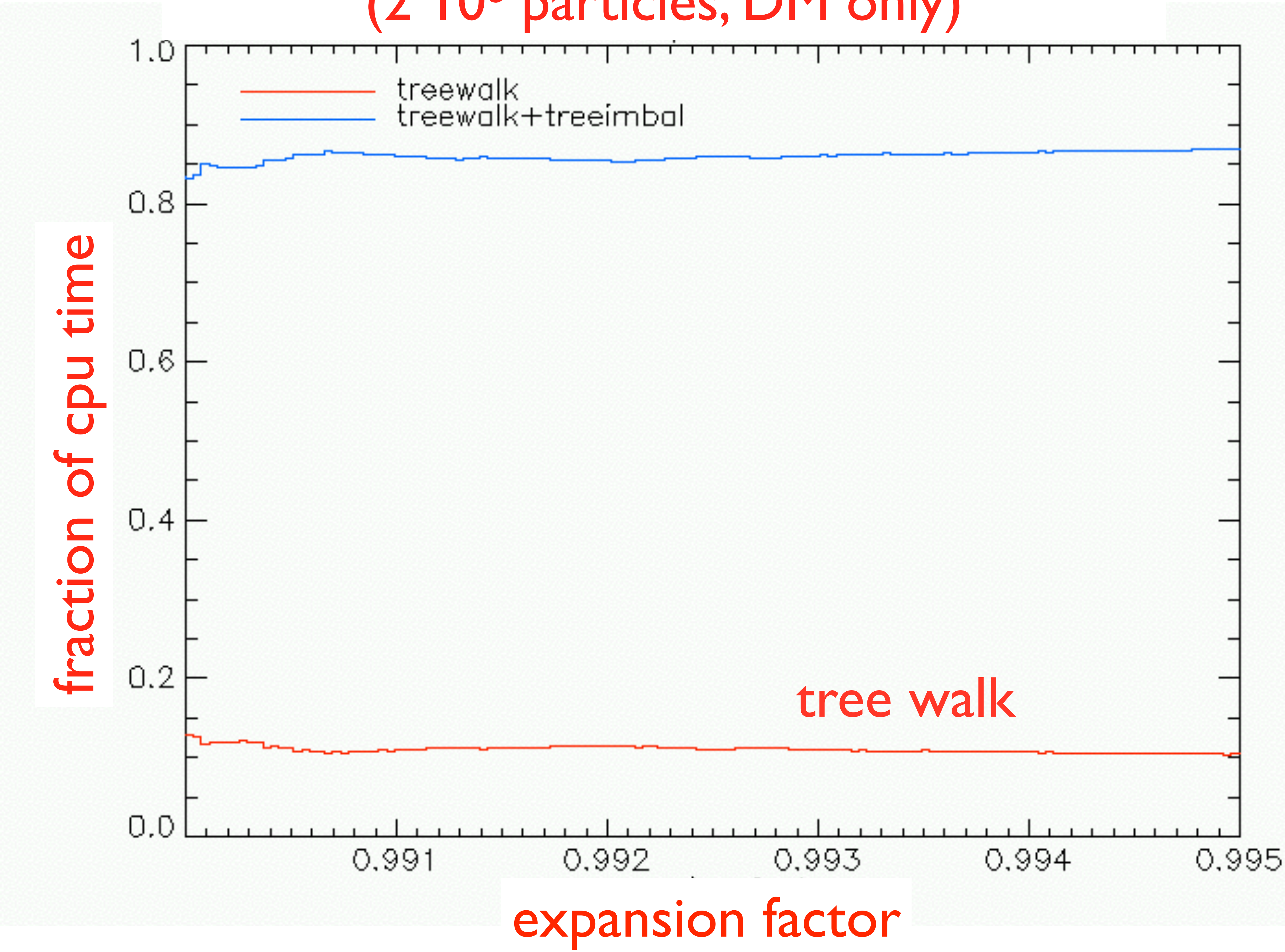
NTask=24
NTask=288
treew
treew+treeimb
treew
treew+treeimb
walk + imbalance (N=288)
walk + imbalance (N=24)
tree walk
total cpu time [s]
expansion factor
1×10^5
8×10^4
6×10^4
4×10^4
2×10^4
0
0.832
0.836
0.837


1×10^5
8×10^4
6×10^4
4×10^4
2×10^4
0
0.832
0.833
0.834
0.835
0.836
0.837
NTask=24
treew
treew+treeimb
NTask=288
treew
treew+treeimb
NTask=600
treew
treew+treeimb
walk + imbalance (N=600)
total cpu time [s]
expansion factor

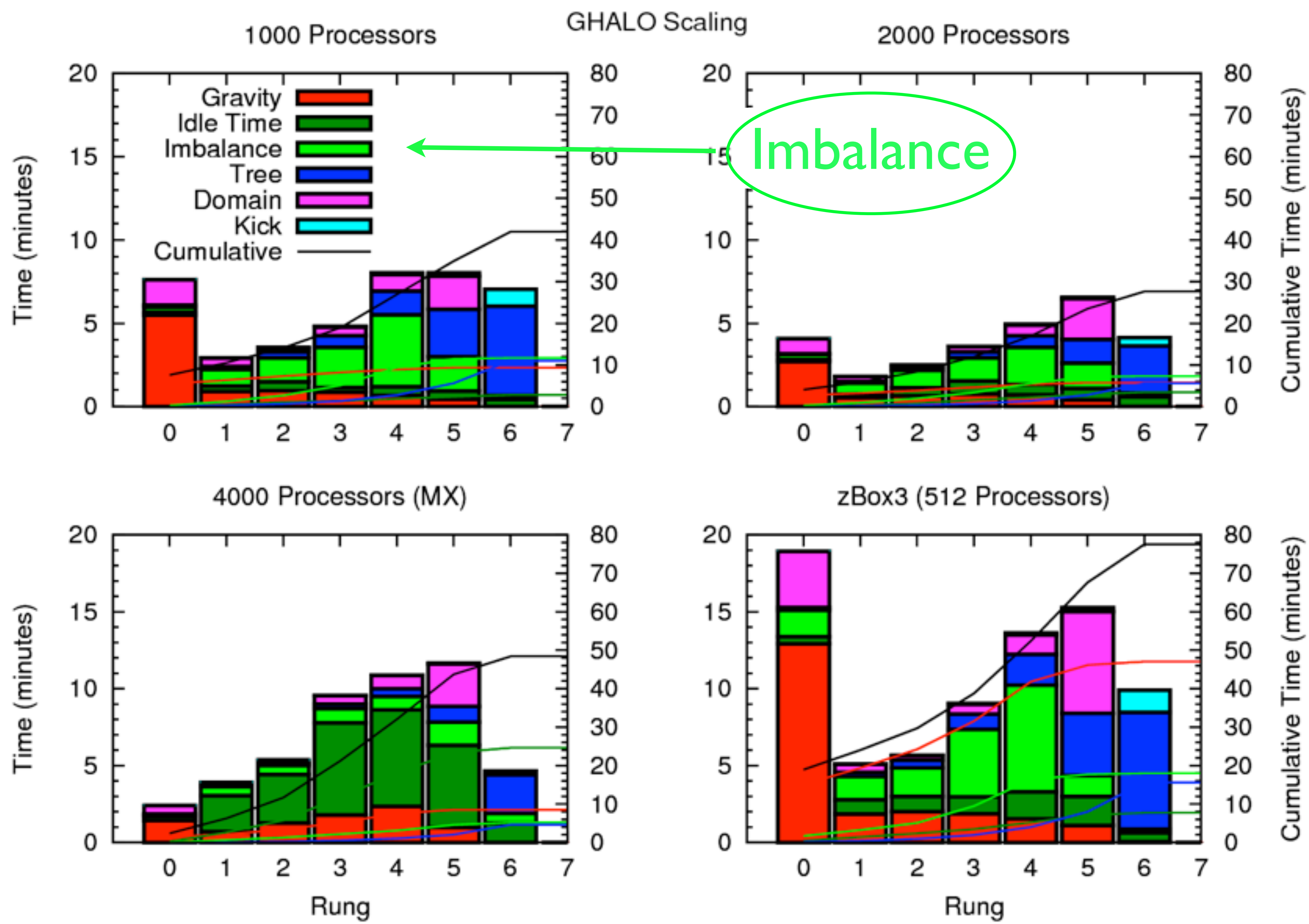
Timings of Gadget-3 on Aquarius Aq-A-2 (2 10^8 particles, DM only)



Timings of Gadget-3 on Aquarius Aq-A-2 (2 10^8 particles, DM only)

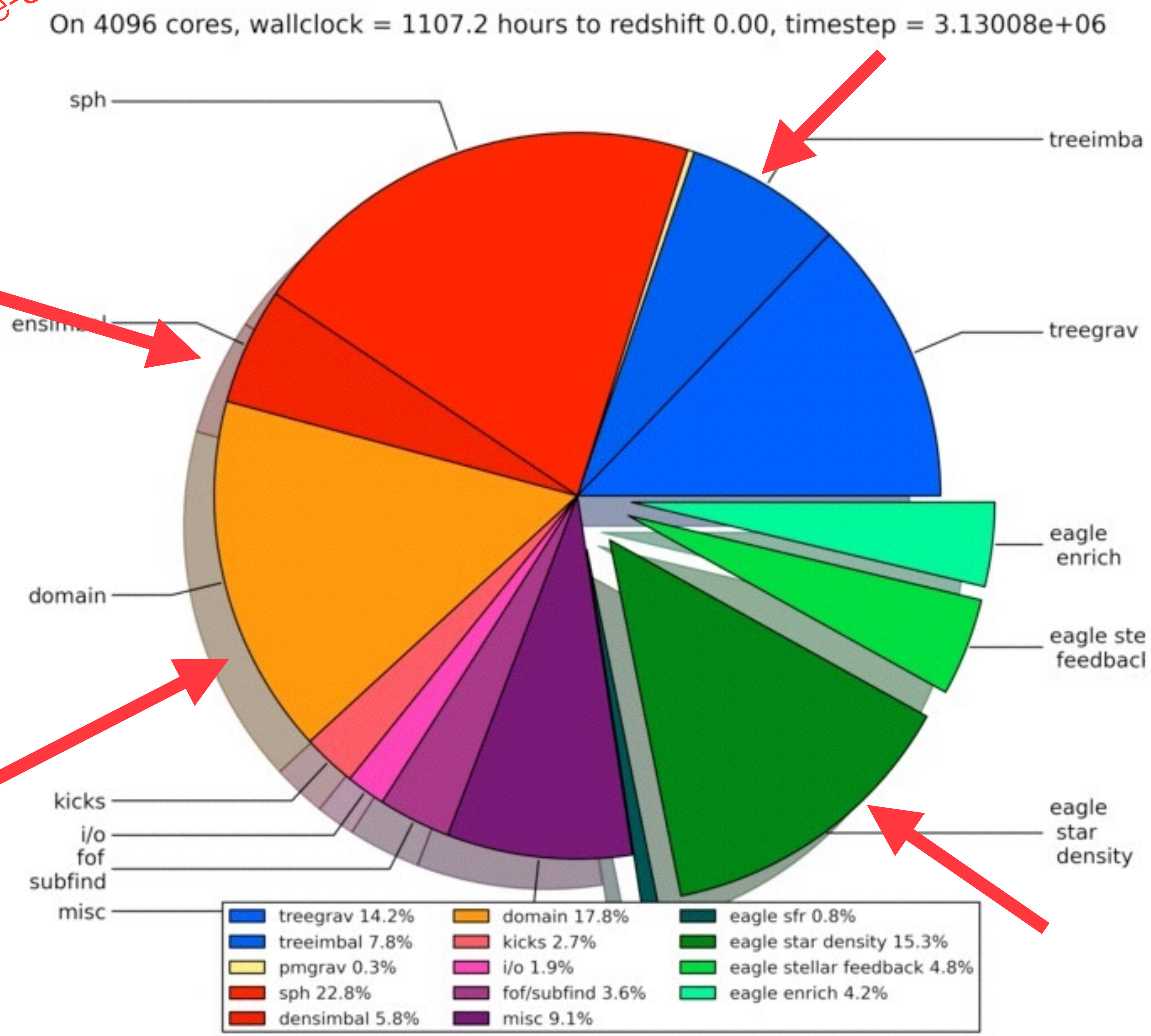


Timings of pkdgrav on G-halo from Stadel



Gadget-3 Tree-SPH code

load imbalance limits scaling



HPC limitations:

tree-walk

```
numngb_inbox = ngb_treefind_variable(&pos[0], h, &startnode);
```

- limited cache efficiency (random data access)
- limited vectorizability

limits single core performance

main integration loop

```
domain()
gravity_tree
density()
hydro_force()
```

- load-imbalance build-up

limits scaling

SPH issues: “blob test”

Fundamental differences between SPH and grid methods

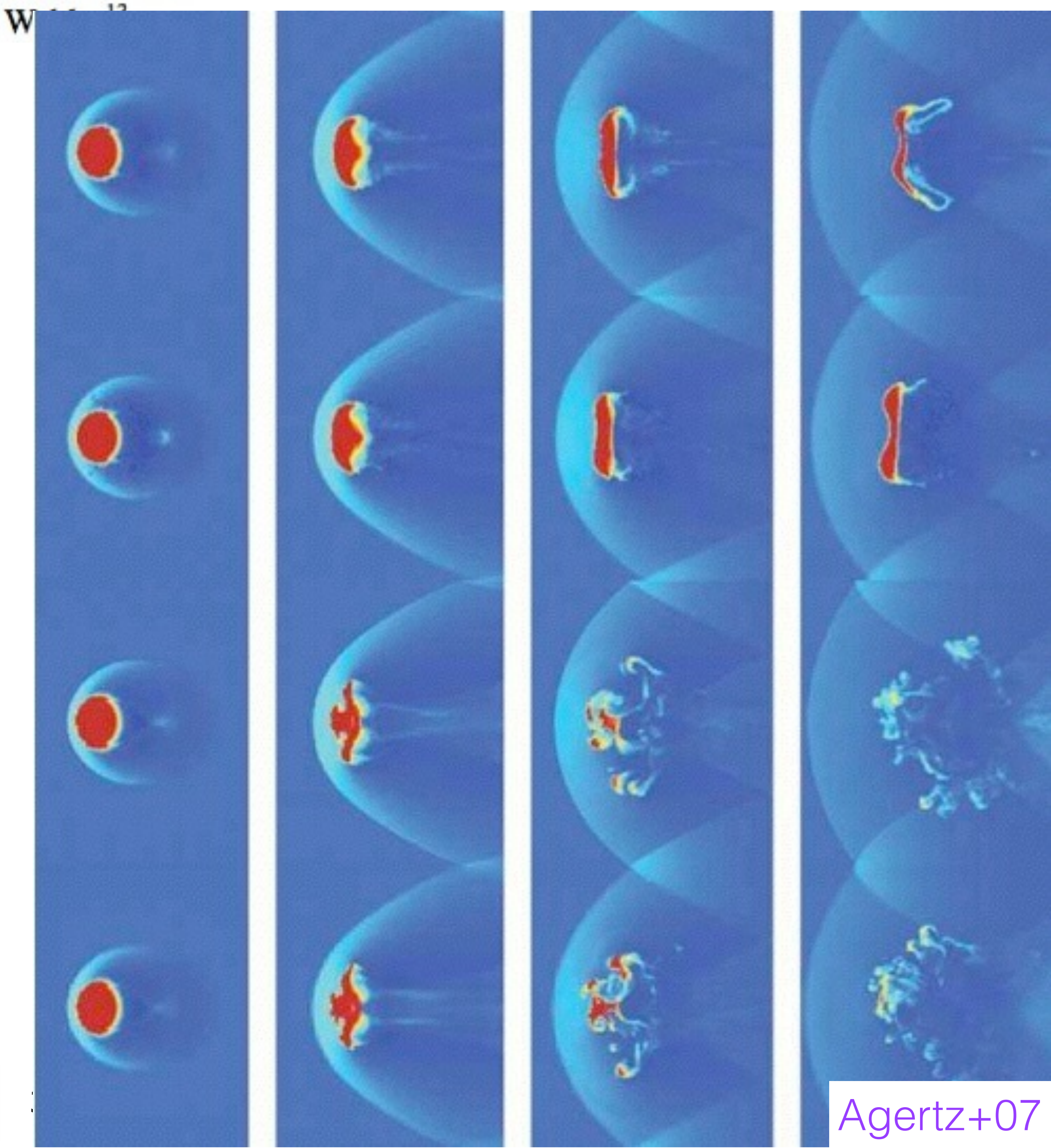
Oscar Agertz,[1★] Ben Moore,[1] Joachim Stadel,[1] Doug Potter,[1] Francesco Miniati,[2]
Justin Read,[1] Lucio Mayer,[2] Artur Gawryszczak,[3] Andrey Kravtsov,[4] Åke Nordlund,[5]
Frazer Pearce,[6] Vicent Quilis,[7] Douglas Rudd,[4] Volker Springel,[8] James Stone,[9]
Elizabeth Tasker,[10] Romain Teyssier,[11] James Wadsley[12] and Rolf W

SPH

SPH

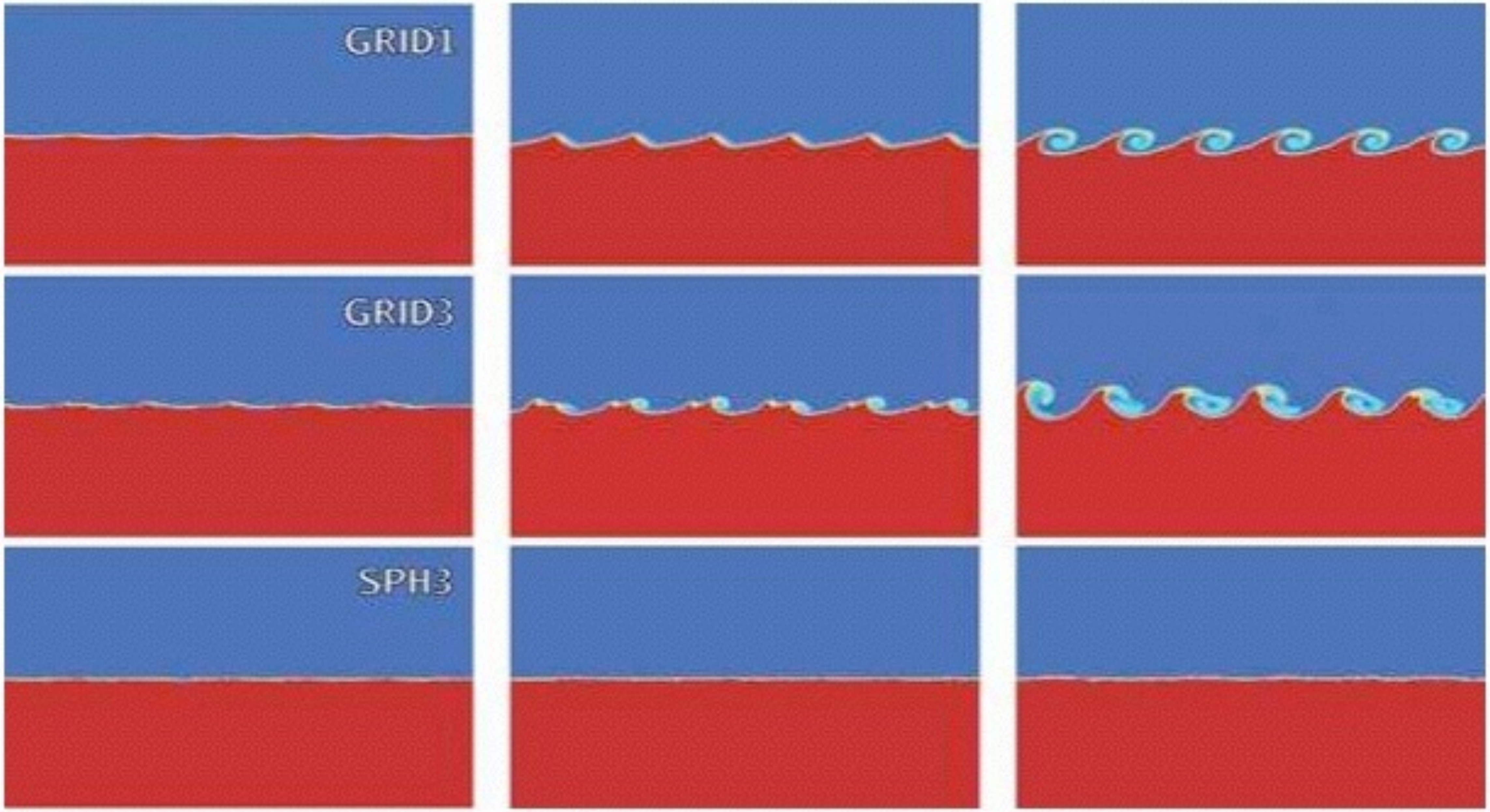
AMR

AMR



Agertz+07

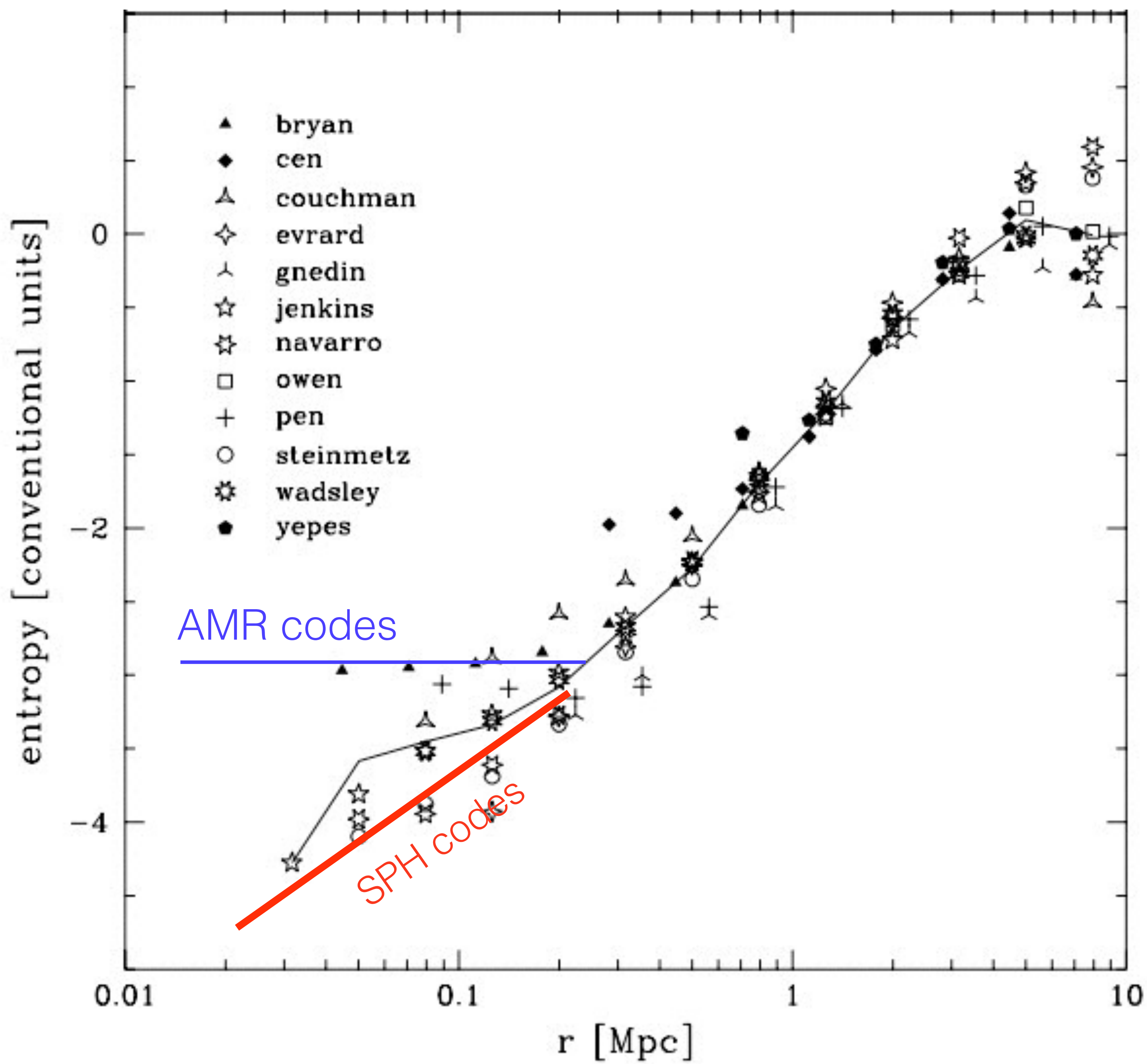
SPH issues: “Kelvin-Helmholtz instability”



Agertz+07

SPH issues: “entropy cusps”

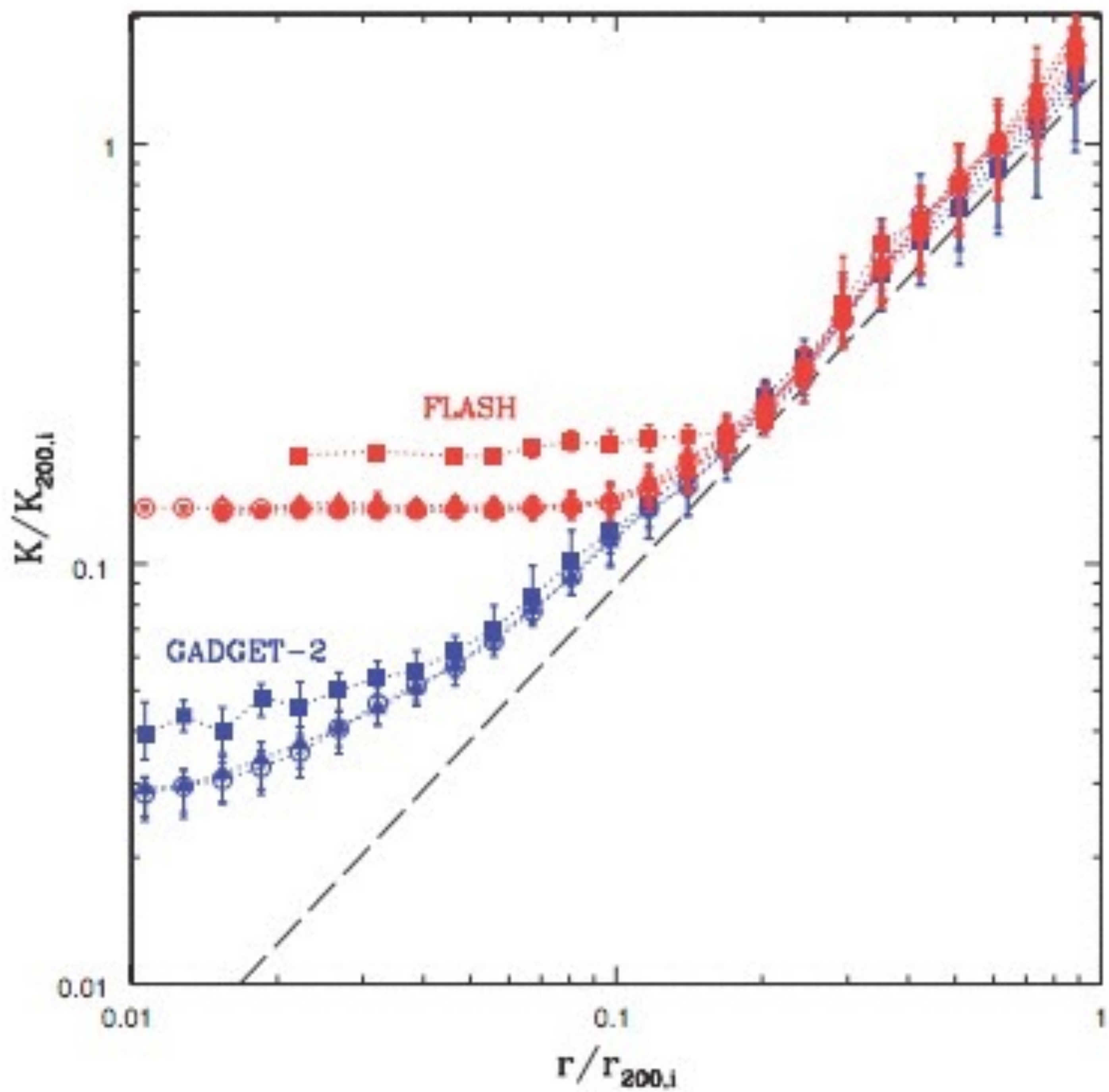
galaxy-cluster entropy profile SPH vs AMR

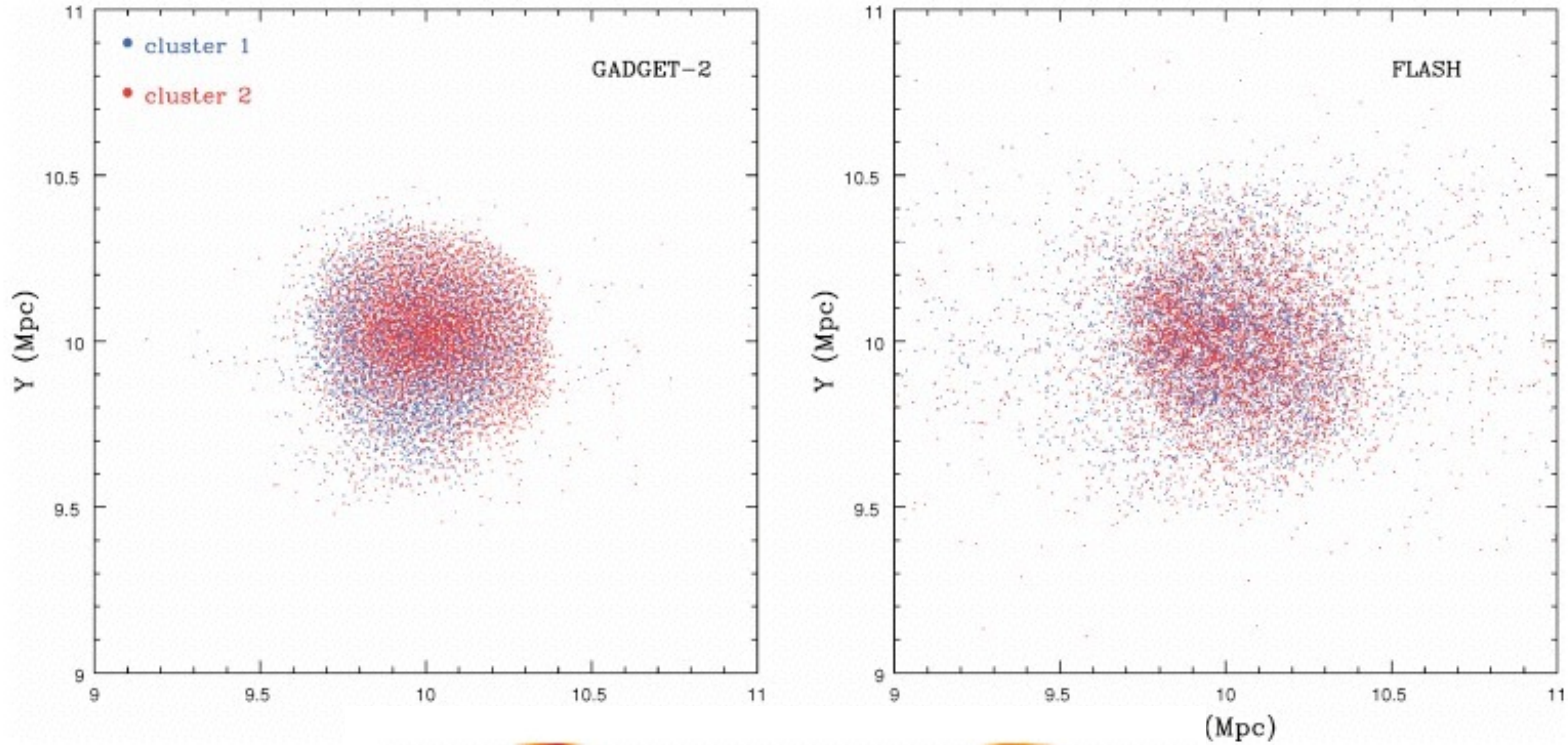


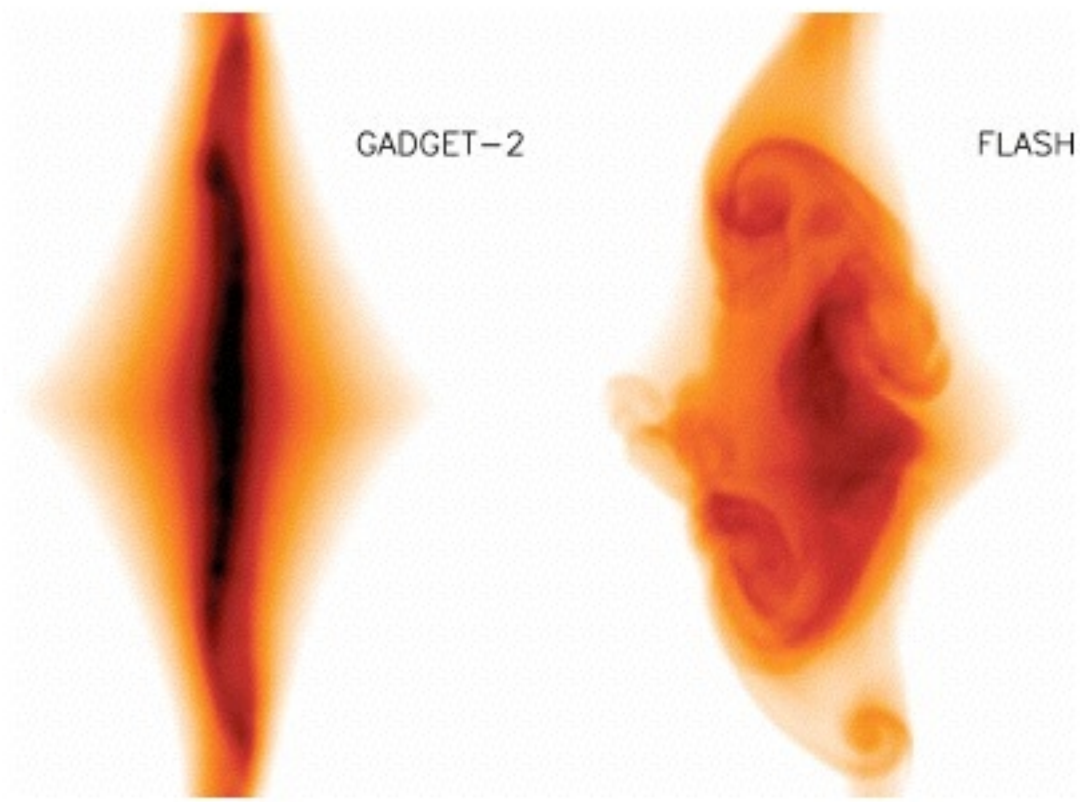
Frenk+99

On the origin of cores in simulated galaxy clusters

N. L. Mitchell,[1★] I. G. McCarthy,[1,2] R. G. Bower,[1] T. Theuns[1,3] and R. A. Crain[1]




cluster 1
cluster 2
GADGET-2
FLASH
Y (Mpc)
(Mpc)
9
9.5
10
10.5
11


GADGET-2
FLASH

The report of SPH's death have been greatly exaggerated

SPH "Anarchy" implementation

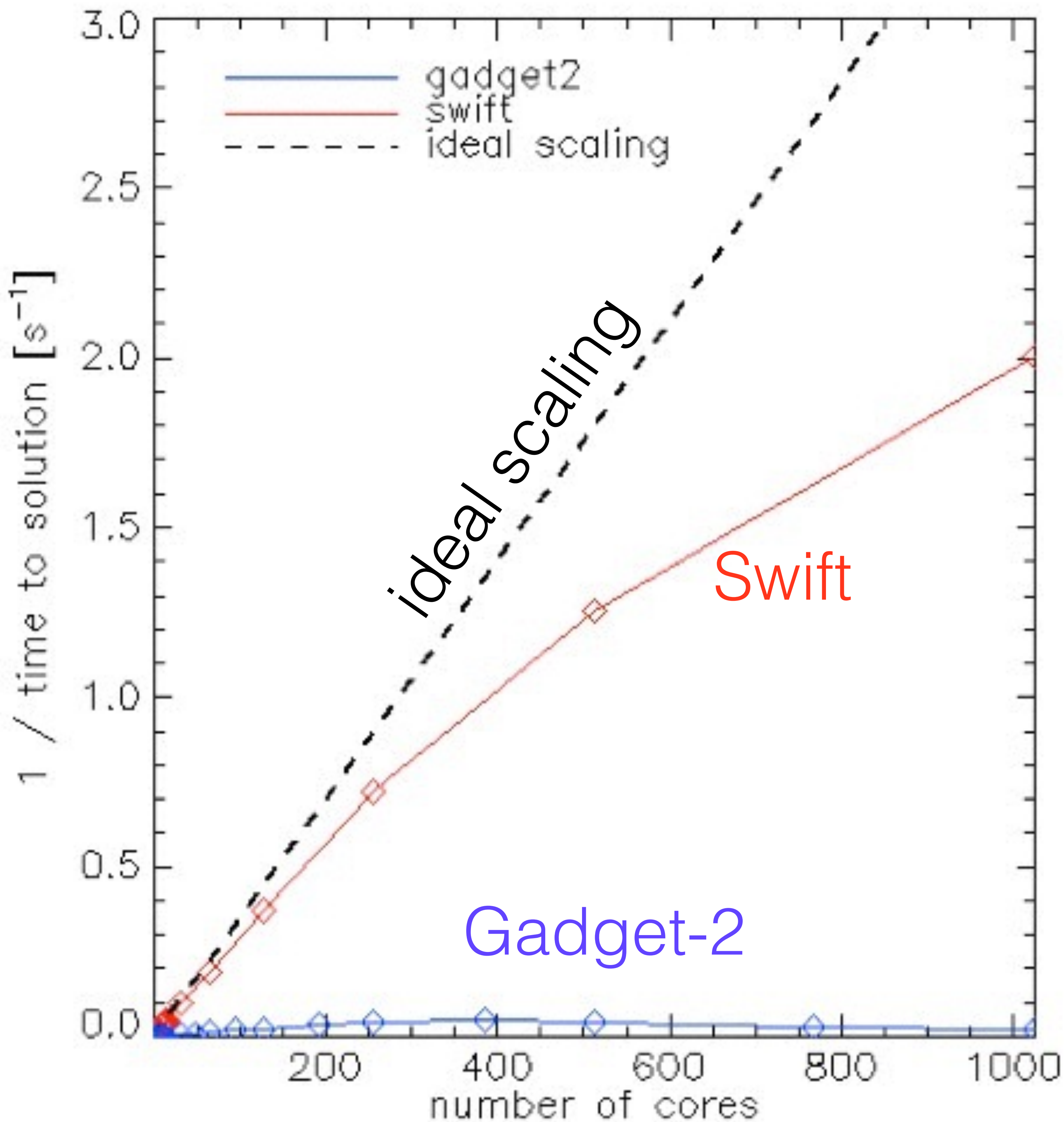
arXiv:1309.3783

SWIFT: Fast algorithms for multi-resolution SPH on multi-core architectures

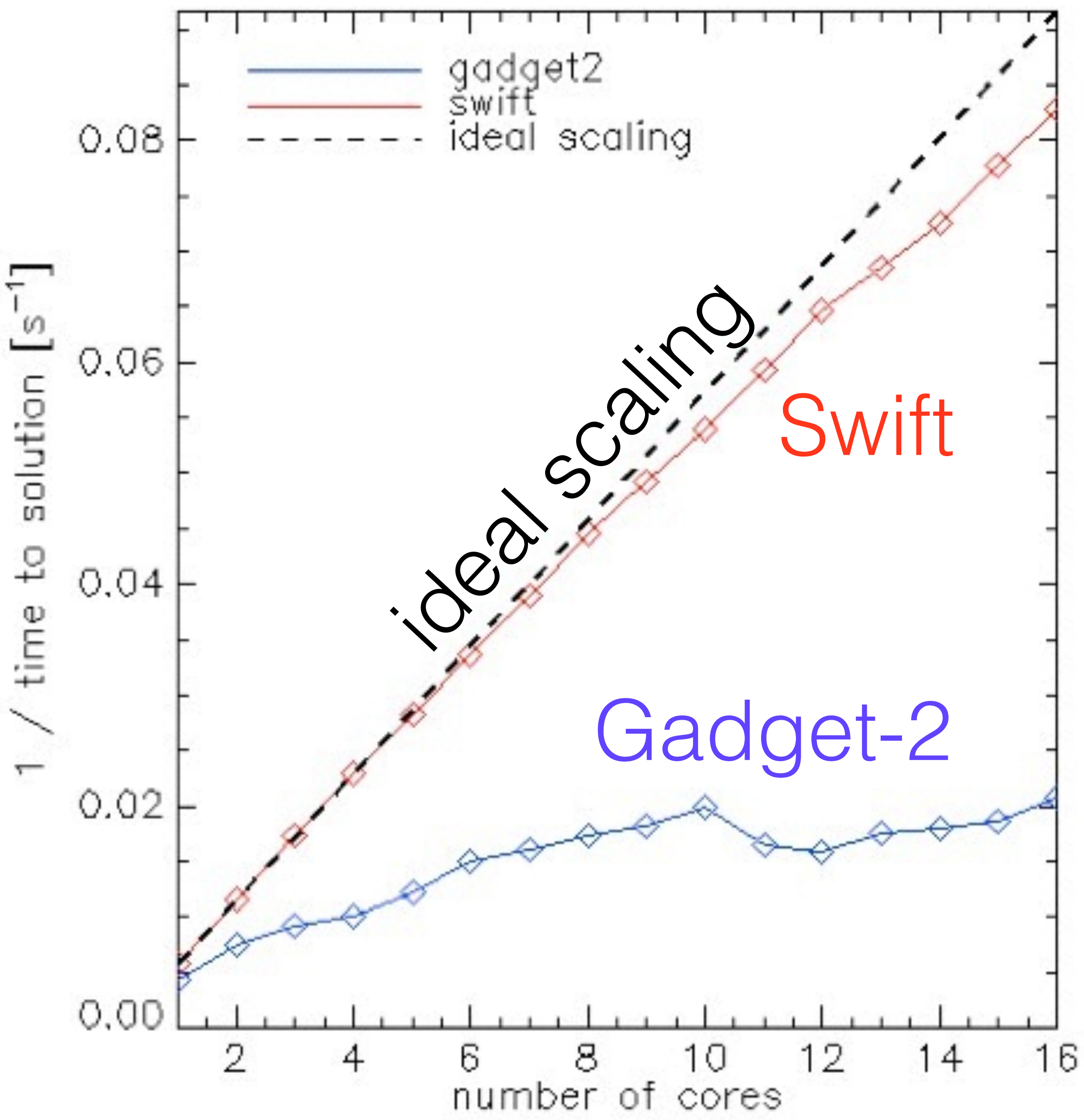
arXiv:1508.00115

Swift: task-based hydrodynamics and gravity for cosmological simulations

Strong scaling test of SPH



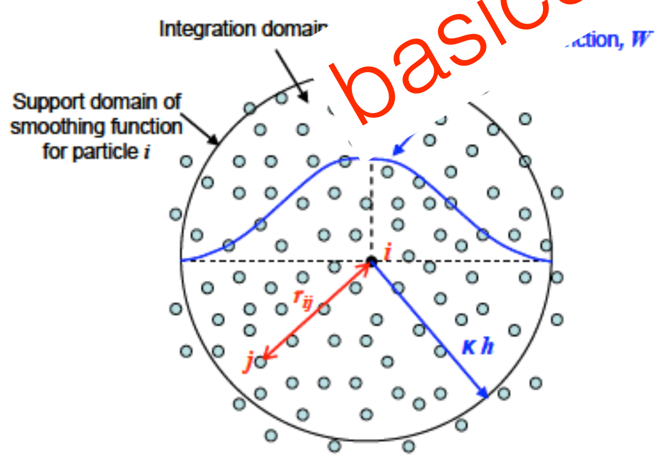
Strong scaling test of Barnes-Hut gravity implementation



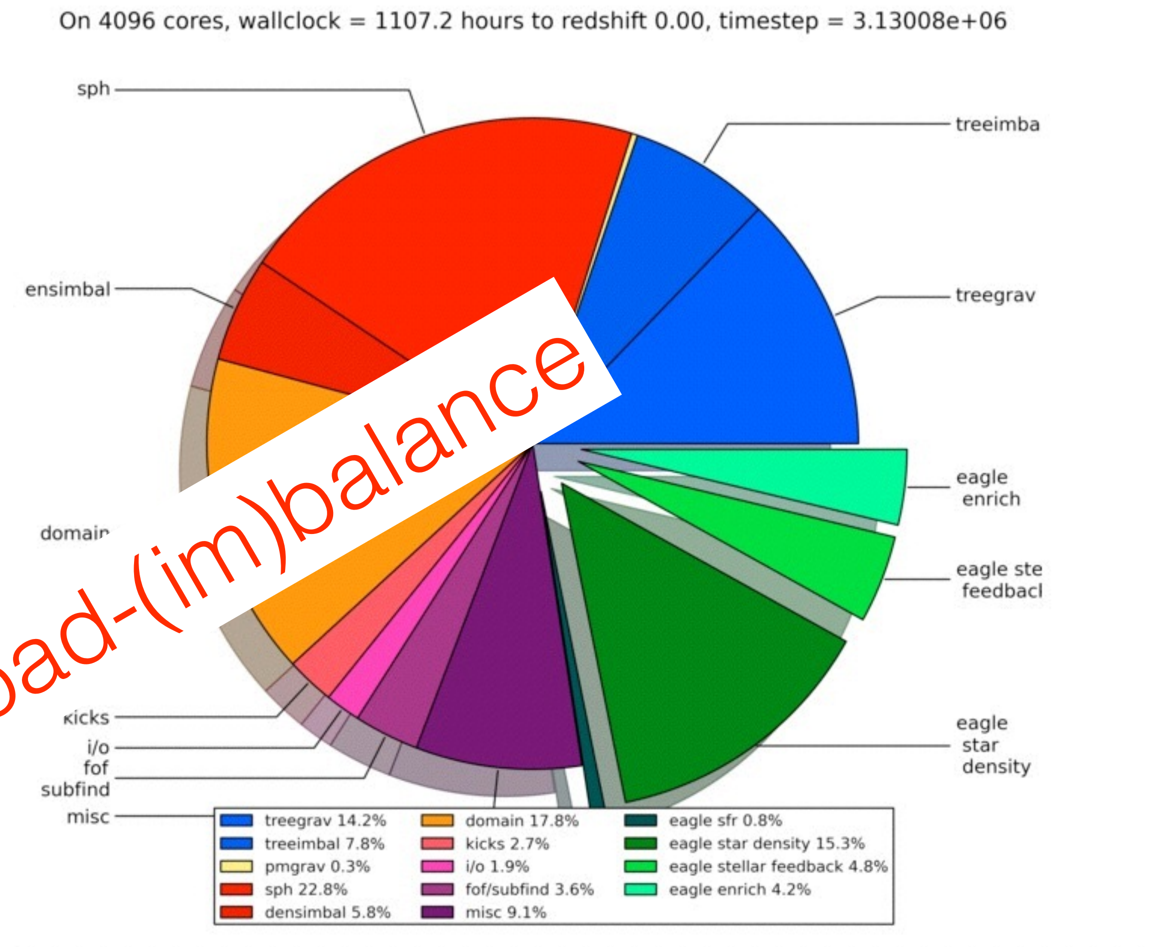
Summary:

$$\rho(\mathbf{r}_i) = \sum_j m_j W\left(\frac{|\mathbf{r}_i - \mathbf{r}_j|}{h_i}\right)$$

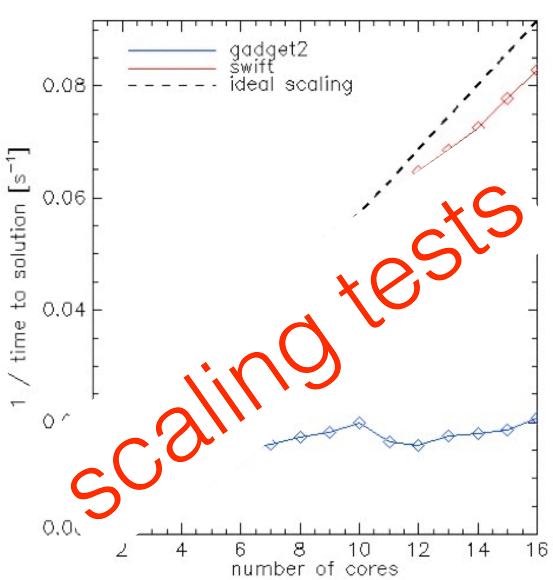
basics



load-(im)balance



scaling tests



issues with standard SPH